

# Finding Digest of Classification Rules

**Abstract**—Here, we address a problem of selecting a representative set of interesting rules inducted from data. We propose a new approach which does not depend on ad hoc thresholds values of some criteria, such as support and confidence. First, we define preference relationship on rules. The single relationship allows comparison of rules by both generality and validity. Then, we define “digest of rules” as a minimal subset of rules, which includes a preferable rule for any rule not in the digest. As a result, the digest comprises the most important and diverse knowledge in data. We propose an algorithm of the search for the digest, show existence and uniqueness of the digest for a given data. The results are obtained for production rules, which generalize such types of rules as association rules and interval rules.

## I. INTRODUCTION

The goal of this work is to present a data-driven approach for the search of compact representative set of probabilistic production rules describing distinctions between two classes in data.

Finding a compact subset of interesting rules is an important problem in data mining. Consider, for example, one of the most popular approaches: association rules [1]. Interesting association rules are defined by the thresholds on support and confidence of the rules. The empirical thresholds are supposed to be selected before the analysis. However, this is not a simple task. Essentially, the level of association of features with outcome on a given dataset is its fundamental characteristic, which needs to be discovered during the analysis.

As an instructive illustration, let us consider a paper [3], where a new way to trim some redundant association rules is proposed. To demonstrate the method, the author applies it on several datasets. The levels of the supports he uses on these datasets are ranging from 0.5%, to 97%. However, the process of the selection of the thresholds is not described. It is obvious that with wrong level of support, the analysis produces meaningless results: either too many or too few rules, some of which are not reliable at all.

Similar problems arise in other data mining approaches. The probabilistic rough set approach proposed in [4] requires setting up two *certainty control* limits for inclusion of elementary sets in the set approximation, as well as *certainty gain* threshold for the positive and boundary regions. The thresholds need to be adjusted for a given dataset.

The paper [2] outlines important problems, associated with exploratory data mining based on the pre-selected thresholds. Particularly, selection of the thresholds before the analysis may lead to lost significant, predictive patterns and abundance of spurious ones. The authors demonstrate it using real datasets, when patterns with high support do not get confirmation on

hold out data, but some patterns with low support turn out to be significant.

The authors [2] proposed selecting certain number of the “best” rules. However, if rules are selected by any validity criterion alone, without taking into account diversity of the patterns, possibility of expressing the dependencies in various terms, necessity of covering most of instances by some rules, we may end up with homogeneous set of rules, which does not serve the purpose of research.

The idea of our approach is to find a minimal combination of rules, called digest, containing the “very best” rules which can be inducted from the data, representing well the variety of the rules on the datasets. The digest defined through the preference relationship on the set of rules, based on the rules validity and generality. We prove existence and uniqueness of the digest for a given dataset.

## II. THE PROBLEM STATEMENT

Consider a two-class inductive learning problem. Suppose the data are given in the form  $[X, Y]$ , where  $X = \{x_{i,j}\}$ ,  $i = 1, \dots, m$ ,  $j = 1, \dots, n$  is matrix with description of  $n$  observations (cases) by  $m$  features;  $Y = \{y_1, \dots, y_n\}$  is a binary outcome vector, assigning a class to each observation. For  $j \in [1, n]$ , if  $y_j = c$ , ( $c \in \{1, 2\}$ ), we say that the  $j$ -th observation  $x_j$  belongs to the class  $c$ . Let  $C_1, C_2$  be all the observations from the classes 1 and 2, respectively.

The goal is to generalize the data and find rules which describe how one class is different from another.

We assume (as it is common in medical applications, for example) the descriptive features represent only part of the factors which affect classification of the objects. Other factors are unknown. The values of the descriptive features as well as classification may be contaminated by noise. Therefore, one can not expect to discover absolutely “true” rules. The validity of the rules is not binary, but a continuous function. It evaluates not only the level of confirmation of the rule on the given data, but its potential to work on new data.

We want to achieve the goal of the problem by finding the most valid rules the data support.

An advantage of the rule-based classification comparing with other approaches is that the inducted rules may be used to understand the dependence between the class and descriptive variables. It is valuable for the data understanding to know how various features can be used to describe the dependence.

Most of times the rules are used to support decision making. When selecting the best set of rules, it is also desirable to increase the variety explained cases.

## A. Language

Below, we consider probabilistic production rules of the form

$$f(x_1, q_1^1, \dots, q_1^k) \& \dots \& f(x_m, q_m^1, \dots, q_m^k) \leftrightarrow (y = i),$$

where  $f(x, q_1, \dots, q_k)$  is a **condition predicate (condition)**, depending on variable  $x$  and  $k$  parameters,  $k$  is a natural number or 0.

Antecedent and consequent of a production rule will be called **clause** and **conclusion**, respectively.

Call condition predicate **trivial**, if it is true everywhere on the feature domain.

We will consider only the production rules where there are conditions on every variable. This does not restrict generality of the approach, since some of these conditions may be trivial. The advantage is that it allows one to code all the rules by the fixed length vectors.

We say the clause  $A$  has length  $l$ , if it has  $l$  nontrivial conditions.

The type of the condition predicates may be selected specifically for each problem. Here are some examples.

- Association rules may be expressed with condition predicate  $f(x, a)$  with one three-valued parameter  $a$ :
  - $a = 1$  means that the item  $x$  is present in the itemset;
  - $a = 0$  means that the item  $x$  is absent in the itemset;
  - $a = -1$  means that the item  $x$  is either present or absent (trivial condition).
- Interval-type rules may be presented using condition predicate with two parameters  $f(x, a, b) \equiv a \leq x \leq b$ . If  $[a, b]$  is the domain of the variable  $x$ , the condition  $f(x, a, b)$  became trivial.

Let us show, how this formalism with interval rules can be used to describe a health - related decision.

Suppose, we have dataset with three features:  $p_1 = \text{cough}$ ,  $p_2 = \text{headache}$  and  $p_3 = \text{temperature}$ . We need to learn to distinguish flu (class 1) from something else. We assume, “cough” and “headache” are nominal features with two values: “yes” and “no”, which are coded by 1 and 0, respectively, in the dataset. Consider the next rule: “If there is no headache, but the temperature is above 100 degrees, this is flu”. Then, the rule may be presented in the next way:

$$(0 \leq p_1 \leq 1) \& (0 \leq p_2 \leq 0) \& (100 \leq p_3 \leq \infty) \leftrightarrow (y = 1).$$

The rule contains trivial condition on the feature “cough”.

## B. Generality

Let us formally define the concept of generality of the clauses. Denote,  $a_i, b_i, i \in \{1, \dots, m\}$  conditions in the clauses  $A, B$  respectively.

**Definition 2.1:** The clause  $A$  is more general than the clause  $B$  :

$$A \supseteq B,$$

if for every  $i, i \in \{1, \dots, m\}$ ,  $b_i \vdash a_i$ .

The generality is a partial order on the clauses. Particularly, if the clauses have different subsets of non-trivial conditions, they are not comparable by generality.

The generality is associated with support and simplicity of rules.

- 1) **If  $A \supseteq B$ , the support of the rule  $A \leftrightarrow c$  is larger or equal to support of the rule  $B \leftrightarrow c$ .** Support of the rule is proportional to the number of instances satisfying the clause of the rule. Indeed, any observation, satisfying the clause  $B$  satisfies the clause  $A : B \models A$ . Therefore, the number of instances satisfying the clause  $A$  is not less than the number of instances satisfying the clause  $B$ .
- 2) **If  $A \supseteq B$ , the clause  $A$  is not longer than the clause  $B$ .** The length of the clause is defined above as number of nontrivial predicates among its conditions. Indeed, if a condition  $i, i \in \{1, \dots, m\}$  in the clause  $B$  is a trivial predicate, then the same condition is a trivial predicate in the clause  $A$  as well. However, it is possible that a condition  $j$  is trivial in the clause  $A$  but not in the  $B$ .

## C. Preference

In this work, we do not discuss possible criteria for rules validity. Assume that some validity criterion is selected based on the requirements of the task at hand.

The concept of the preference relationship on rules integrates both generality and validity.

**Definition 2.2:** For given validity criterion  $Q$ , for rules  $A$  and  $B$  with the same conclusion, we say that rule  $A$  is **preferable** to rule  $B$  :

$$A \gg_Q B,$$

if

- 1) The rules  $A$  and  $B$  are comparable by generality;
- 2) Either

$$Q(A) > Q(B)$$

or

$$(Q(A) = Q(B)) \& (A \supseteq B).$$

According with this definition, rules not comparable by generality can not be comparable by preference. We will skip the index  $Q$  from the notation of the preference relationship, when it will not cause confusion.

Selection of rules by this criterion will preserve variety of descriptions of rules, which is one of the goals of learning process.

At the same time, it will help the classification of widest possible variety of cases. Indeed, if rules are not comparable by generality, each rule covers some cases, which are not explained by another rule. Then, neither rule will be considered preferable to another one.

If the rules are comparable by generality, the generality itself is used only to break ties by validity to compare rules by preference. This is because we assume the validity measures not only confirmation of the rule on the given data, but also its potential to be expanded on new cases.

Let us notice that preference **is not a transitive** relationship. For example, the rule

(a1) *umbrellas & wet pavement*  $\hookrightarrow$  *it is rain*

is less general, but, perhaps, more reliable than the rule

(a2) *umbrellas*  $\hookrightarrow$  *it is rain*.

Sometimes people carry umbrellas to defend themselves from hot sun. Therefore, we can assume that the validity of rule (a1) is higher. Then, (a1) is preferable to (a2):  $(a1) \gg (a2)$ . If the rule

(a3) *wet pavement*  $\hookrightarrow$  *it is rain*

is as valid as rule (a1), the rule (a3) is preferable to the rule (a1):  $(a3) \gg (a1)$ , because it is more general. So,  $(a3) \gg (a1)$ ,  $(a1) \gg (a2)$ , yet, the rules (a2) and (a3) are not comparable by preference since they are not comparable by generality.

#### D. Digest of Rules

Instead of defining the “best” individual rules, we want to define a subset of rules which, as a whole, includes “the most important” knowledge one can induce from the data. This idea is captured in the concept of the digest of rules defined below.

Denote  $R(c)$  set of all syntactically correct production rules with the conclusion  $y = c$ .

*Definition 2.3:* For a given validity function  $Q$ , the set of rules  $D \subseteq R(c)$  is a **digest of the class  $c$**  if

1) For every rule  $P \in (R(c) \setminus D)$  there exists a rule  $T \in D$  :  $T \gg P$ ;

2) Any two rules in  $D$  are not comparable by preference.

The first property means that digest contains a preferable rule for any rule outside of the digest.

The second property describes digest as a minimal subset of rules satisfying the first property: non of the rules in the digest is “better” than any other rule in the digest, therefore no rule can be excluded without violation of the first property.

But does the set with such properties exist, and if yes, is it unique? To answer these questions, we introduce a “concentration algorithm”, and show that the algorithm always finds a digest and this is the only digest for given data.

### III. CONCENTRATION ALGORITHM

#### A. Definition

*Definition 3.1:* Given is the validity function  $Q$ , the class  $c$ . First, the pool of candidate rules  $G = R(c)$ ; the set of rules  $D$  is empty.

1) Find the set  $T$  of all the rules with the highest validity in  $G$ ;

2) Find subset  $\hat{T}$  of all the most general rules in  $T$ , include it in  $D$  :  $D = D \cup \hat{T}$ .

3) Exclude all rules comparable by generality with  $\hat{T}$  from the pool  $G$ .

4) If the set  $G$  is not empty, repeat the steps 1 – 3 above.

*Theorem 3.1:* After the concentration algorithm stops, the set  $D$  is the digest of the rules for a class  $c$ ; and there is no other digest of rules.

**Proof.** We call the sequence of the steps 1 – 3 an **iteration**. The algorithm consists of repeating the iterations, until the pool  $G$  is empty.

Let us notice that on each iteration validity of rules added into digest is lower than on the previous iterations. Every time, on the step 3, the rules  $\hat{T}$  are removed from the pool  $G$  together with all rules comparable with  $\hat{T}$  by generality. This includes all the set  $T$  with all the rules with highest validity in the current pool  $G$ .

We need to prove that **(a)** rules in  $D$  are not comparable by preference; **(b)** for any rule not in  $D$ , there is a preferable rule in  $D$ ; **(c)**  $D$  is the only set of clauses with such properties.

**(a).** Suppose, the rule  $d$  is added on the iteration  $k$ . The rule  $d$  does not have a preferable rule added on the same iteration, because on the step 2 we take care that only the most general rules out of the selected candidates  $T$  will be added to digest. The rule  $d$  can not have preferable rules added on the previous iterations, because in each previous iteration  $j$ ,  $j < k$  on the step 3 we exclude from the search space  $G$  all the rules, comparable by generality with the rules added on the iteration  $j$ . Since all the rules added before are not comparable by generality with  $d$ , they are not comparable by preference. The rule  $d$  does not have preferable rules added on the iteration  $i$  :  $i > k$ , since all the rules added on the next iterations have lower validity.

**(b).** Suppose, the rule  $d$  is not in  $D$ . It means, it was excluded from the search space  $G$  on some of the iteration on the step 3. The rules excluded on the step 3, are less preferable than some rules added to the digest on the same iteration.

**(c).** We need to show that rules added into digest belong to every digest, and rules excluded from the pool of candidates  $G$  do not belong to any digest. Let us show it by induction on the count of iterations, on which rules are added to digest or excluded from the pool  $G$ .

Rules, which are added to the digest in the first iteration, can not have preferred rules in any digest, because they have highest validity in the set of all rules, and they are the most general among the rules with highest validity. Therefore, these rules belong to every digest.

The rules, which were excluded from the pool of candidates  $G$  on the first iteration, can not be in any digest, because they are less general than some of the rules with the same or higher validity, present in every digest.

Suppose, the statement is proven for all iterations up to  $(k - 1)$ . Let us show it for the iteration  $k$ .

Consider a rule  $A$ , selected to be in a digest on the iteration  $k$ . As we noticed, the rules with higher validity are dealt with on the previous iterations. Every one of them, by the induction statement, is either in every digest, or is excluded from the pool of candidates for any digest on the previous iterations. Therefore, there will not be a preferable rule with higher validity in any digest.

Since the rule  $A$  is included in a digest on the iteration  $k$ ,

$A$  is among the most general rules with the same validity in the pool  $G$  of the iteration  $k$ . It means each rule with the same validity either belongs to  $G$  and is less preferable than  $A$ , or it is excluded from  $G$  on some previous iteration. The rules excluded from  $G$  on iterations up to  $k - 1$  can not belong to any digest by the induction statement. Therefore,  $A$  does not have a preferable rule with the same validity in any digest.

Since  $A$  can not have a preferable rule in any digest either with the same validity, or with higher validity,  $A$  has to belong to every digest by the definition of digest. This shows that rules added to a digest on the iteration  $k$  belong to every digest.

Rules, excluded from the pool of candidates in the iteration  $k$ , are less preferable than some of the rules, which added on this iteration and belong to every digest, as we demonstrated. Therefore, they do not belong to any digest.

#### Q.E.D.

Since the digest of rules of a class is unique for given validity function, being in the digest for a certain class is a property of a rule.

### B. Properties of the digest

Let us notice some important properties of the concept of the digest and the concentration algorithm.

- 1) Even though rules in the digest do not have preferable ones in the digest, they may have (and some of them do have) preferable rules outside of the digest. For example, suppose, the rules with itemsets  $z_1, z_2$  have the same validity  $v_1$ , highest for the current set of transactions, and  $z_1 \subset z_2$ . Then, the rule with the itemset  $z_1$  is more general than the rule with the itemset  $z_2$ , and it is preferable. The digest will contain the rule with itemset  $z_1$ , and not the rule with itemset  $z_2$ . Now, suppose, the rule with the itemset  $z_3 \subset z_2$  has validity  $v_2 : v_2 < v_1$ . Then, the rule with the itemset  $z_2$  is preferable to the rule with the itemset  $z_3$ . Yet, the rule with the itemset  $z_3$  may belong to the digest.
- 2) The concentration algorithm finds rules starting from the most valid ones. If the process takes too long to complete, it may be interrupted, when there is enough rules. The obtained set of rules will contain all highest validity rules supported by the given data.

### C. Implementation of the Concentration Algorithm

Below, we describe useful shortcuts, which make the concentration algorithm feasible.

For this description, we will need a concept of chain of clauses.

**Definition 3.2:** Chain of clauses is a maximal sequence of clauses  $\{C_1, \dots, C_k\}$  such that for any  $1 \leq i \leq k - 1$ ,

$$C_i \supseteq C_{i+1}.$$

According with the mentioned first property of the generality relationship on the clauses, support of the clauses along the chain monotonically decreases.

We make two assumptions:

- 1) All set of the clauses is broken into sequence of chains  $\xi_1, \dots, \xi_r$ .

- 2) The algorithm evaluates clauses on each chain in the order of clauses on the chain, and consider all chains in their fixed order from  $\xi_1$  to  $\xi_r$ .

**1) Simplifying the Step 1: How to skip ends of chains.** Suppose, there is a “bottom” level of support  $s$  for a clause, such that clauses with support below  $s$  are not considered worthy attention. Then when the algorithm encountered a clause with support below  $s$ , it can drop evaluating clauses on the current chain and switch to the next chain in the order of chains. It may significantly decrease the time for the step 1, when it performed the first time.

**How to skip chains.** Let us show that we can skip more and more chains with each iteration.

Suppose, a clause  $A$  from a chain  $\xi_j$  was added to digest on the iteration  $k$ . Then, any other clause from this chain can not be added to digest on any next iteration.

To show it, consider a clause  $B \neq A$  from the same chain  $\xi_j$ . By definition of a chain,  $A, B$  are comparable by generality.

If  $B$  was not included in  $T_k$ , it is less valid than  $A$ . Therefore  $A \gg B$ . If  $B$  was included in  $T_k$  but did not get into digest,  $A \supseteq B$ , otherwise the clause  $A$  would be excluded from  $T_k$ . Then, again,  $A \gg B$ . By definition of digest,  $B$  can not belong to the digest.

This means that on each iteration we can skip all the chains, from which some clauses were already added to digest on previous iterations.

**2) Simplifying the step 3:** The step 3 requires to exclude from the pool of clauses all the clauses, comparable by generality with ones, added to the digest on an iteration  $i$ . Doing it literally is impractical. Instead, one may check for each of clauses being added in an iteration  $i$ , if the digest already contains more general clause.

## IV. APPLICATION OF THE CONCENTRATION ALGORITHM

To demonstrate advantages of the digest of rules, we applied the algorithm on the real life dataset along with more traditional approach, which finds rules based on the thresholds of the support and confidence.

The data represent information about prostate cancer patients from Memorial Sloan-Kettering Cancer Center. The goal is to predict clinical failure of a patient (death or metastases) during 5 years after prostatectomy. The patients are characterized by 17 features. The information includes clinical characteristics, some measurements of abundance of androgen receptor in the prostate tissue, as well as histological properties of the tissue, identified during computerized analysis of photo images of H & E stained prostate tissue. The dataset was randomly split on the training and test data. The training set has 295 records, the test set has 288 records. The clinical failure class makes only 7.8% of the training set, and only 5.9% of the test set. Most of features are continuous.

On the training data, we applied the concentration algorithm, as well as the Live Logic procedure, [5] which finds the most general interval rules, satisfying given constraints on support and confidence. With both algorithms, we built only rules with not more than two conditions in their premises.

For the Live Logic procedure, we used 98% and 20% as thresholds for the confidence and support, respectively. The threshold for the confidence was selected based on the proportion of the first class in data: confidence shall be higher than proportion of the largest class, but less than 100%. The threshold for the support was selected after the first run of the algorithm with the support threshold 10% generated too many rules. The algorithm found 6 rules for the first class (clinical failure) and 72 rules for the second class. All rules for the first class were confirmed on the test dataset, having lift more than 5.56 on the test data. The rules for the second class have lift on the training equal 1.06; on the test the lift ranges from 0.97 to 1.04. In all cases, the rules for the second class are useless for practice, since the condition of the rule improves chances of good outcome very little.

The concentration algorithm was used with  $z$ -test as validity measure. It was conditioned to find not more than 75 best rules total. The algorithm found 58 rules, all of them for the poor outcome class. All the rules had  $p$ -value by  $z$ -test below 0.001 both on the training and on the test set.

The example demonstrates important advantages of digest of rules:

- 1) One does not need to have “preliminary” runs or “guess” upfront which thresholds need to be chosen for given data, given class.
- 2) Unlike the threshold - based approach, the concentration algorithm produced statistically significant rules not only on the training, but on the hold out data as well.
- 3) Concentration algorithm produced much more predictive rules, confirmed on the test data. Some of these rules do not satisfy the selected for the second approach thresholds for support and confidence, yet all of them give conditions for much higher incidence of poor outcome than on whole dataset. This is an important result for predicting the outcome of the surgery.
- 4) Concentration algorithm allowed us us to eliminate all the useless rules obtained based on the selected high thresholds of support and confidence.

## V. CONCLUSIONS

- 1) We propose a method for a search of compact set of rules, representing in certain sense all the variety of rules which can be learned from data.
- 2) This subset of rules is defined without setting any subjective thresholds or limits on interestingness of rules.
- 3) We propose an algorithm of search for digest of rules for each class.
- 4) We show that the digest of rules always exists and it is unique.
- 5) We applied the algorithm on real life medical dataset, to compare it with more traditional approach for selection of interesting rules. All rules were tested on the hold out data. We demonstrated that the proposed concentration algorithm allowed to find much more valuable rules, and to skip useless ones.

## REFERENCES

- [1] Agrawal, R., Mannila, H., Strikant, R., Toivonen, H. and Verkamo, A.I. Fast discovery of association rules. In Fayyad, U. M. et al editors, *Advances in Knowledge Discovery and Data Mining* (Cambridge, MA: AAAI Press, 1996) 307–328.
- [2] Webb G, Zhang, S. K-Optimal Rule Discovery. *Data Mining and Knowledge Discovery*, 10,2005, 39-79.
- [3] Zaki M J, Hsiao C-J. Efficient Algorithms for Mining Closed Itemsets and Their Lattice Structure, *IEEE Transaction on Knowledge and Data Engineering*, 17(4), 2005, 462-478.
- [4] Ziarko, W. 2005. Probabilistic Rough Sets. *Rough sets, fuzzy sets, data mining, and granular computing*. Lecture notes in artificial intelligence N 3641. Dominik Slezak et al (eds). (Springer, Berlin, 2005).
- [5] Sapir M, Verbel D, Kotsianti A, Saidi O (2005) Live Logic: Method for Approximate Knowledge Discovery and Decision Making. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, 10th International Conference, RSFDGrC 2005, Regina, Canada, August 31 - September 3, 2005, Proceedings, Part I. Lecture Notes in Computer Science 3641, 532-540