2014 22nd International Conference on Pattern Recognition

VC-Dimension of Rule Sets

Olcay Taner Yıldız
Department of
Computer Engineering
Işık University
Istanbul
Email: olcaytaner@isikun.edu.tr

***Abstract*—In this paper, we give and prove lower bounds of the VC-dimension of the rule set hypothesis class where the input features are binary or continuous. The VC-dimension of the rule set depends on the VC-dimension values of its rules and the number of inputs.**

***Index Terms*—VC-Dimension, Rule sets**

I. Introduction

Rule induction, an old branch of machine learning, is concentrated on extracting rule sets from data. A rule set is typically composed of an ordered list of rules, where a rule is composed of a conjunction of a list of conditions [1]. Depending on the type of the input attribute, the conditions are of the form

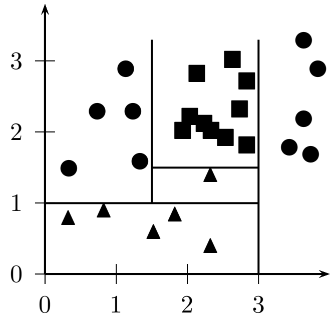
- $x_i = v$: if x_i is discrete
- $x_i \leq \theta$ or $x_i > \theta$: if x_i is continuous

A rule is said to *cover* an instance, if that instance satisfies all conditions in that rule. Each rule is associated with a class label, and class label of the first covering rule is assigned to an instance. If none of the rules cover an instance, the default class label is assigned to that instance. An example rule set composed of three rules is given below.

```
If $x_2$ = 0 and $x_4$ = 0 Then $C_1$
Else
    If $x_1$ = 1 and $x_3$ = 0 and $x_5$ = 1 Then $C_1$
    Else
        If $x_1$ = 0 Then $C_1$
        Else $C_0$
```

Well known rule induction algorithms are C4.5Rules [2], PART [3], CN2 [4] and Ripper [5]. C4.5Rules generates a decision tree and then transforms it to a set of rules by writing each path from the root to a leaf as a rule; PART grows a partial decision tree and extracts a single rule from the best performing leaf; Ripper and CN2 directly produce a set of rules.

There are two main groups of rule induction algorithms: Separate-and-conquer algorithms and divide-and-conquer algorithms. This paper is mainly related with the algorithms following separate and conquer strategy [1]. According to this strategy, when a rule is learned for class C_i, the covered examples are removed from the training set. This procedure proceeds until no examples remain from class C_i in the training set. If we have two classes, we separate positive class from negative class. But if we have $K > 2$ classes, as a heuristic, every class is classified in the order of their increasing prior probabilities, i.e., in the order of their sample size.



```
If ($x_1$ < 3) and ($x_2$ < 1)
  Then class = ▲
Else
  If ($x_1$ > 1.5) and ($x_2$ < 1.5)
    Then class = ▲
  Else
    If ($x_1$ < 1.5)
      Then class = ●
    Else
      If ($x_1$ > 3)
        Then class = ●
      Else class = ■
```

Fig. 1. For a specific class ordering, separation of data and learned ruleset.

Ripper, as an example of separate-and-conquer strategy of algorithms, learns rules to separate a positive class from a negative class. In Figure I we see an example case, where Ripper first learn rules to separate class ▲ from both classes ● and ■, then learn rules to separate class ● from class ■.

Vapnik-Chervonenkis (VC) dimension is a measure of complexity defined for any hypothesis class, that is, class of functions [6]. VC dimension for a class of functions $f(\mathbf{x}, \alpha)$, where α denotes the parameter vector, is defined to be the largest number of points that can be shattered by members of $f(\mathbf{x}, \alpha)$. A set of data points is *shattered* by a class of functions $f(\mathbf{x}, \alpha)$ if for all assignments of class labels to those points, one can find a member of $f(\mathbf{x}, \alpha)$ which makes no errors

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.615

when evaluating that set of data points. For example, in two dimensions, we can separate three points with a line, but we can not separate four points (if the assignments of class labels are done like in the famous XOR problem). Therefore, the VC dimension of the linear estimator class in two dimensions is 3.

In this work, we use rule sets with binary or continuous input attributes as our hypothesis classes. In this case, α corresponds to a vector of condition counts, representing the number of conditions in each rule. For example, the rule set given above is an element of the hypothesis class $f(\mathbf{x}, [2, 3, 1]_3)$, where the first, second, and third rules have 2, 3, and 1 conditions respectively.

As far as our knowledge, there is no explicit formula for the VC-dimension of a rule set. On the contrary, there are certain results for the VC-dimension of decision trees [7], [8], [9]. These bounds are either structure independent, that is, they give the same bound for all decision trees with N nodes; or the bounds are for particular type of univariate trees. In our previous work, we proved structure dependent lower bounds of the VC-dimension of univariate decision trees with binary features [10].

Our approach is the following: First, for four basic rule set structures, we give and prove a lower bound of the VC-dimension. Second, we give and prove a general lower bound of the VC-dimension of the rule set with binary features. Third, based on those theorems, we give an algorithm to find a structure dependent lower bound of the VC-dimension of a rule set with binary features. As a last step, we generalize our work to include continuous data, that is continuous rule set hypothesis class. We again give an algorithm to find a lower bound of the VC-dimension of a rule set for continuous data sets.

Note that we are discussing the VC dimension of hypotheses classes defined as families of rule set that share the rule set structure and differ only in the variables being tested in the internal decision conditions.

This paper is organized as follows: In Section II, we give and prove the lower bounds of the VC-dimension of the rule sets with binary features. We generalize our work to continuous rule sets in Section III and conclude in Section IV.

II. VC-dimension of the Rule Sets with Binary Features

We consider the well-known supervised learning setting where the rule set algorithm uses a sample of m labeled points $S = (\mathbf{X}, \mathbf{Y}) = ((\mathbf{x^{(1)}}, y^{(1)}), \ldots, (\mathbf{x^{(m)}}, y^{(m)}) \in (\mathcal{X} \times \mathcal{Y})^m$, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ the label set, which is $\{0, 1\}$. The input space $\mathcal{X}$ is a vectorial space of dimension d, the number of features, where each feature can take values from $\{0, 1\}$.

Each rule set $\mathcal{R}$ is represented with a vector $\mathcal{R} = [r_1, r_2, \ldots, r_k]_k$, where there are r_1, r_2, ..., r_k conditions in the first, second, ..., k^{th} rule respectively. Note again that we are searching the VC dimension of rule set hypotheses class that share the rule set structure and differ only in the function being tested in the decision conditions.

Theorem 1: The VC-dimension of rule set $\mathcal{R}_1 = [1]_1$ (a single rule composed of a single decision condition) that classifies d dimensional binary data is $\lfloor \log_2(d+1) \rfloor + 1$.

$$\mathbf{X} = \begin{array}{cccccccc} & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 \\ \mathbf{x}^{(1)} & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ \mathbf{x}^{(2)} & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ \mathbf{x}^{(3)} & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ \mathbf{x}^{(4)} & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}$$

If $x_5 = 1$ **Then** C_1 **If** $x_3 = 1$ **Then** C_1
Else C_0 **Else** C_0

Fig. 2. Example for Theorem 1 with $d = 7$ and $m = 4$. If the class labeling of S is {1, 1, 0, 0} we select feature 5 (left rule set). If the class labeling of S is {0, 0, 1, 0} we select feature 3 (right rule set).

Proof: To show the VC-dimension of the rule set $\mathcal{R}_1$ is at least m, we need to find such a sample S of size m that, for each possible class labelings of these m points, there is an instantiation h of our hypothesis class $\mathcal{R}_1$ that classifies it correctly. Let $\mathbf{C_m}$ be the matrix of size $2^{m-1} - 1 \times m$ which represents all possible division of m data points into two classes. For m = 4, the matrix $\mathbf{C_4}$ is

$$\mathbf{C_4} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

We construct the sample S such that

$$\mathbf{X} = \mathbf{C_m^T}$$

that is, each feature d_i corresponds to a distinct possible class labeling of m points, implying a one-to-one mapping between class labelings and features. So for each possible class labeling, we will choose the rule set hypothesis h which has the corresponding feature as the feature in the condition (See Figure 2 for an example).

A sample with m examples can be divided into two classes in $2^{m-1} - 1$ different ways. If we set the number of features to that number:

$$\begin{aligned} d &= 2^{m-1} - 1 \\ d + 1 &= 2^{m-1} \\ \log_2(d+1) &= m - 1 \\ m &= \log_2(d+1) + 1 \end{aligned}$$

To show the VC-dimension of rule set $\mathcal{R}_1$ is at most $\lfloor \log_2(d+1) \rfloor + 1$, we go in reverse direction. If the VC-dimension of the rule set $\mathcal{R}_1$ is m, for each possible class combination of m examples, we must be able to separate them into two classes. In the rule set $\mathcal{R}_1$, we can use at most d possible orthogonal splits. A sample with m examples can be divided into two classes in $2^{m-1}-1$ different ways. In order to be able to separate m instances for each possible class combination, the total number of splits must be at least as large as the number of possible class divisions. So,

$$\begin{aligned} d &\geq 2^{m-1}-1 \\ m &\leq \log_2(d+1)+1 \end{aligned}$$

■

Theorem 2: The VC-dimension of of rule set $\mathcal{R}_2 = [h]_1$ (a single rule composed of a h decision conditions) that classifies d dimensional binary data is $\left\lfloor \log_2\left(\binom{d}{h}+1\right)\right\rfloor+1$.

Proof: The proof is similar to the proof of Theorem 1. We only give the reverse direction. If the VC-dimension of $\mathcal{R}_2$ is m, for each possible class combination of m examples, we must be able to separate them into two classes. In $\mathcal{R}_2$, we can have at most $\binom{d}{h}$ possible orthogonal splits corresponding to selected h of d features. A sample with m examples can be divided into two classes in $2^{m-1}-1$ different ways. In order to be able to separate m instances for each possible class combination, the total number of splits must be at least as large as the number of possible class divisions. So,

$$\begin{aligned} \binom{d}{h} &\geq 2^{m-1}-1 \\ m &\leq \log_2\left(\binom{d}{h}+1\right)+1 \end{aligned}$$

■

Theorem 3: The VC-dimension of rule set $\mathcal{R}_3 = [1,1,\ldots,1]_k$ that classifies d dimensional binary data is at least $\lfloor \log_2(d-k+2)\rfloor+k$.

Proof: Similar to the Theorem 1, we produce a sample S such that, for each possible class labelings of this sample, there is an instantiation h of our hypothesis class $\mathcal{R}_3$ that classifies the sample correctly. We construct the sample S such that

$$\mathbf{X} = \begin{bmatrix} \mathbf{C_m^T} & \mathbf{0} \\ \mathbf{0} & I_{k-1} \end{bmatrix}$$

where I_{k-1} shows the identity matrix of size $k-1 \times k-1$.

Here, we proceed in a bottom-up fashion. The bottom node can classify m examples by setting up $2^{m-1}-1$ features to produce a one-to-one mapping between class labelings and those features (See Theorem 1). We also add one feature and one example for each remaining node, where the value of the new feature is 1 for the corresponding example and 0 for the remaining examples (See Figure 3 for an example). The classification of the sample goes as follows: $k-1$ rules (which have a single decision condition) will select the new $k-1$ features respectively as the features in their decision conditions so that each added example will be covered by that rule alone. The remaining m examples will be forwarded to the bottom rule, where that rule and else part can classify those examples whatever their class combination is. The number of features is,

$$\begin{aligned} d &= 2^{m-1}-1+k-1 \\ d-k+2 &= 2^{m-1} \\ m &= \log_2(d-k+2)+1 \end{aligned}$$

So the VC-dimension of the hypothesis class $\mathcal{R}_3$ is at least $(m+k-1)$, that is $\lfloor \log_2(d-k+2)\rfloor+k$. ■

$$\mathbf{X} = \begin{array}{c|ccccccc} & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 \\ \hline \mathbf{x}^{(1)} & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ \mathbf{x}^{(2)} & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ \mathbf{x}^{(3)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{x}^{(4)} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \mathbf{x}^{(5)} & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \mathbf{x}^{(6)} & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \mathbf{x}^{(7)} & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$$

```
If $x_7 = 1$ Then $C_x$
Else
    If $x_6 = 1$ Then $C_x$
    Else
        If $x_5 = 1$ Then $C_x$
        Else
            If $x_4 = 1$ Then $C_x$
            Else
                If $x_3 = 0$ Then $C_0$
                Else $C_1$
```

Fig. 3. Example for Theorem 3 with $d = 7$ and $m = 7$. If the class labeling of S is $\{1, 1, 0, x, x, x, x\}$ we select feature 3 in the bottom rule. The labelings of the last four examples do not matter since they are alone in the rules they reside.

Theorem 4: The VC-dimension of rule set $\mathcal{R}_4 = [h,h,\ldots,h]_{2^{h-1}}$ that classifies d dimensional binary data is at least $2^{h-1}(\lfloor \log_2(d-h+2)\rfloor+1)$.

Proof: Let $\mathbf{B_{x,m,d}}$ be the matrix of size $m \times d$ which contains m identical rows of binary representation of integer x. For example, the matrix $\mathbf{B_{6,5,3}}$ is

$$\mathbf{B_{6,5,3}} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$$

We construct the sample S such that

$$\mathbf{X} = \begin{bmatrix} \mathbf{C_m^T} & \mathbf{B_{0,m,h-1}} \\ \mathbf{C_m^T} & \mathbf{B_{1,m,h-1}} \\ \ldots & \ldots \\ \mathbf{C_m^T} & \mathbf{B_{2^{h-1}-1,m,h-1}} \end{bmatrix}$$

Each of 2^{h-1} rules,

(I) contains a possible combination of values of $h-1$ features (See Figure 4 for an example), and

$$
\mathbf{X} = \begin{array}{c|ccccc}
 & d_1 & d_2 & d_3 & d_4 & d_5 \\
\mathbf{x}^{(1)} & 1 & 0 & 1 & 0 & 0 \\
\mathbf{x}^{(2)} & 0 & 1 & 1 & 0 & 0 \\
\mathbf{x}^{(3)} & 0 & 0 & 0 & 0 & 0 \\
\mathbf{x}^{(4)} & 1 & 0 & 1 & 0 & 1 \\
\mathbf{x}^{(5)} & 0 & 1 & 1 & 0 & 1 \\
\mathbf{x}^{(6)} & 0 & 0 & 0 & 0 & 1 \\
\mathbf{x}^{(7)} & 1 & 0 & 1 & 1 & 0 \\
\mathbf{x}^{(8)} & 0 & 1 & 1 & 1 & 0 \\
\mathbf{x}^{(9)} & 0 & 0 & 0 & 1 & 0 \\
\mathbf{x}^{(10)} & 1 & 0 & 1 & 1 & 1 \\
\mathbf{x}^{(11)} & 0 & 1 & 1 & 1 & 1 \\
\mathbf{x}^{(12)} & 0 & 0 & 0 & 1 & 1
\end{array}
$$

If $x_4 = 0$ **and** $x_5 = 0$ **and** $x_3 = 0$ **Then** C_0
Else
  If $x_4 = 0$ **and** $x_5 = 1$ **and** $x_1 = 0$ **Then** C_0
  Else
    If $x_4 = 1$ **and** $x_5 = 0$ **and** $x_2 = 0$ **Then** C_0
    Else
      If $x_4 = 1$ **and** $x_5 = 1$ **and** $x_1 = 0$ **Then** C_0
      Else C_1

Fig. 4. Example for Theorem 4 with d = 5 and m = 12. Using features 4 and 5 as the first two features in all rules, one divides the class labelings into 4 subproblems of m = 3. Each subproblem can then be shattered with a single condition. For the example rule, the class labeling of S is {1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0}.

(II) can classify m examples by setting up $2^{m-1} - 1$ features to produce a one-to-one mapping between class labelings and those features (See Theorem 1).

This way each rule can be labeled as a $h - 1$ digit binary number (I) and can shatter m examples by selecting appropriate feature and value in the last condition (II). The number of features is,

$$
\begin{aligned}
d &= 2^{m-1} - 1 + h - 1 \\
d - h + 2 &= 2^{m-1} \\
m &= \log_2(d - h + 2) + 1
\end{aligned}
$$

So the VC-dimension of the rule set $\mathcal{R}_4$ is at least $2^{h-1}m$, that is $2^{h-1}(\lfloor \log_2(d - h + 2) \rfloor + 1)$. ■

Theorem 5: The VC-dimension of a rule set with binary features that classifies d dimensional binary data is at least the maximum of the sum of the VC-dimensions of its sub rule sets those classifying $d - 1$ dimensional binary data.

Proof: Let the VC-dimension of two rule sets ($\mathcal{R}_a = [r_1, r_2, \ldots, r_k]_k$ and $\mathcal{R}_b = [s_1, s_2, \ldots, s_l]_l$) be VC_a and VC_b respectively. Under this assumption, those rule sets can classify VC_a and VC_b examples under all possible class labelings of those examples. Now we form the following rule set: We add a new feature f to the dataset and use that feature as the last condition of each rule in both rulesets $\mathcal{R}_a = [r_1 + 1, r_2 + 1, \ldots, r_k + 1]_k$ and $\mathcal{R}_b = [s_1 + 1, s_2 + 1, \ldots, s_l + 1]_l$. The conditions that will be added to the rule sets $\mathcal{R}_a$ and $\mathcal{R}_b$ will be of the form $x_f = 0$ and $x_f = 1$ respectively. Then we combine the new rules of rulesets $\mathcal{R}_a$ and $\mathcal{R}_b$ to form the new rule set $\mathcal{R}_c = [r_1 + 1, r_2 + 1, \ldots, r_k + 1, s_1 + 1, s_2 + 1, \ldots, s_l + 1]_{k+l}$, where rules of rule set $\mathcal{R}_b$ follow the rules of rule set $\mathcal{R}_a$. Now the rule set can classify at least $VC_a + VC_b$ examples for all possible class labelings of those examples.

```
int VC-Dimension1($\mathcal{R} = [r_1, r_2, \ldots, r_k]_k$, $d$)
if $k$ = 1 and $r_1$ = 1
    return $\lfloor \log_2(d+1) \rfloor + 1$
if $k$ = 1 and $r_1$ != 1
    return $\left\lfloor \log_2 \left( \binom{d}{r_1} + 1 \right) \right\rfloor + 1$
max = 0
for $i$ = 1 to $k$ - 1
    s = VC-Dimension1($[r_1 - 1, \ldots, r_i - 1]$, $d - 1$) +
        VC-Dimension1($[r_{i+1} - 1, \ldots, r_k - 1]$, $d - 1$)
    if s > max
        max = s
return max
```

Fig. 5. The pseudocode of the recursive algorithm for finding a lower bound of the VC-dimension of a rule set for binary data: $\mathcal{R}$: Rule set hypothesis class, d: Number of inputs

Using the above idea, the VC-dimension of rule set $\mathcal{R}_5 = [r_1, r_2, \ldots, r_k]_k$ will be the maximum of the sum of the VC-dimensions of rule sets

- $\mathcal{R}_a = [r_1 - 1]_1$ and $\mathcal{R}_b = [r_2 - 1, \ldots, r_k - 1]_{k-1}$
- $\mathcal{R}_a = [r_1 - 1, r_2 - 1]_2$ and $\mathcal{R}_b = [r_3 - 1, \ldots, r_k - 1]_{k-2}$
- $\ldots$
- $\mathcal{R}_a = [r_1 - 1, \ldots, r_{k-2} - 1]_{k-2}$ and $\mathcal{R}_b = [r_{k-1} - 1, r_k - 1]_2$
- $\mathcal{R}_a = [r_1 - 1, \ldots, r_{k-1} - 1]_{k-1}$ and $\mathcal{R}_b = [r_k - 1]_1$

■

Figure 5 shows the recursive algorithm that calculates a lower bound for the VC-dimension of an arbitrary rule set using Theorems 1, 2 and 5. There are two base cases; (i) the rule set contains only one rule of one condition (Theorem 1), (ii) the rule set contains one rule of r_1 conditions (Theorem 2).

III. VC-Dimension of Rule sets with Continuous Features

Until now, we considered the VC-dimension of rule sets with binary attributes. In this section, we generalize our idea to rule sets with continuous features. For this case, the input space X is a vectorial space of dimension d, where each feature d_i can take values from continuous real space. We assume that, for at least one feature d_i, all instances have distinct values.

Corollary 1: The VC-dimension of rule set $\mathcal{R}_1 = [1]_1$ that classifies d dimensional continuous data is at least $\lfloor \log_2(d + 1) \rfloor + 1$.

Proof: The proof directly follows the proof of Theorem 1 given a slight modification. We construct the sample S such

$$
\mathbf{X} = \begin{array}{cccccccc}
 & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 & d_7 \\
\mathbf{x}^{(1)} & 1.3 & 0.2 & 0.1 & 0.4 & 1.7 & 1.4 & 1.7 \\
\mathbf{x}^{(2)} & 0.9 & 1.3 & 0.7 & 0.1 & 1.1 & 0.1 & 0.1 \\
\mathbf{x}^{(3)} & 0.5 & 0.8 & 1.4 & 0.9 & 0.6 & 1.8 & 0.6 \\
\mathbf{x}^{(4)} & 0.6 & 0.6 & 0.3 & 1.8 & 0.3 & 0.3 & 1.8
\end{array}
$$

If $x_5 \leq 1$ **Then** C_0 **Else** C_1

If $x_3 \leq 1$ **Then** C_0 **Else** C_1

Fig. 6. Example for Corollary 1 with $d = 7$ and $m = 4$. If the class labeling of S is {1, 1, 0, 0} we select feature 5 and the split $x_5 \leq 1$ (left rule set). If the class labeling of S is {0, 0, 1, 0} we select feature 3 and the split $x_3 \leq 1$ (right rule set).

that

$$\mathbf{X} = \mathbf{C}_\mathbf{m}^\mathbf{T} + \mathbf{R}_\mathbf{m}$$

where $\mathbf{R}_\mathbf{m}$ is a random matrix of size $m \times 2^{m-1} - 1$ containing random values from the interval (0, 1). Given such an $\mathbf{X}$, $\left\lfloor x_i^{(t)} \right\rfloor$ will correspond to a possible class labeling of $\mathbf{x}^{(t)}$, implying a one-to-one mapping between class labelings and features. So for each possible class labeling, we will choose the rule set hypothesis h which has the corresponding feature as the split feature and the split is $x_i \leq 1$ (See Figure 6 for an example). ■

Corollary 2: The VC-dimension of rule set $\mathcal{R}_2 = [h]_1$ that classifies d dimensional continuous data is at least $\left\lfloor \log_2 \left(\binom{d}{h} + 1 \right) \right\rfloor + 1$.

Proof: The proof directly directly follows the proof of Theorem 2 with the same modification in Corollary 1. ■

Figure 7 shows the recursive algorithm that calculates a lower bound for the VC-dimension of an arbitrary rule set for continuous data using Corollary 2. We spare one feature and two conditions for dividing the instances into subproblems. The remaining features and conditions are used to separate the instances in each subproblem. We set the values of the spared feature in increasing order from up to down, that is, the instances forwarded to the uppermost/lowermost rule will have the smallest/largest value in that spared feature (See Figure 8 for an example). For this reason, when we encounter a rule with r_i condition in a binary data set, the VC-dimension is $\left\lfloor \log_2 \left(\binom{d}{r_i} + 1 \right) \right\rfloor + 1$, where d represents the remaining features for that rule, whereas when we encounter a rule with r_i condition in a continuous data set, the VC-dimension is $\left\lfloor \log_2 \left(\binom{d-1}{r_i-2} + 1 \right) \right\rfloor + 1$, where d represents the number of all features in that data set.

```
int VC-Dimension2($\mathcal{R} = [r_1, r_2, \ldots, r_k]_k$)
sum = 0
for i = 1 to k do
    sum += $\left\lfloor \log_2 \left( \binom{d-1}{r_i-2} + 1 \right) \right\rfloor + 1$
return sum
```

Fig. 7. The pseudocode of the algorithm for finding a lower bound of the VC-dimension of a rule set for continuous data: $\mathcal{R}$: Rule set hypothesis class

$$
\mathbf{X} = \begin{array}{ccccc}
 & d_1 & d_2 & d_3 & d_4 \\
\mathbf{x}^{(1)} & 1.3 & 0.5 & 1.3 & 0.2 \\
\mathbf{x}^{(2)} & 0.8 & 1.5 & 1.6 & 0.3 \\
\mathbf{x}^{(3)} & 0.7 & 0.8 & 0.4 & 0.2 \\
\mathbf{x}^{(4)} & 1.4 & 0.3 & 1.2 & 0.7 \\
\mathbf{x}^{(5)} & 0.6 & 1.3 & 1.5 & 0.9 \\
\mathbf{x}^{(6)} & 0.5 & 0.5 & 0.6 & 0.7 \\
\mathbf{x}^{(7)} & 1.3 & 0.6 & 1.5 & 1.2 \\
\mathbf{x}^{(8)} & 0.7 & 1.6 & 1.3 & 1.4 \\
\mathbf{x}^{(9)} & 0.9 & 0.2 & 0.7 & 1.3 \\
\mathbf{x}^{(10)} & 1.4 & 0.7 & 1.3 & 1.7 \\
\mathbf{x}^{(11)} & 0.7 & 1.6 & 1.6 & 1.8 \\
\mathbf{x}^{(12)} & 0.6 & 0.8 & 0.3 & 1.6
\end{array}
$$

If $x_4 > 0$ **and** $x_4 \leq 0.5$ **and** $x_3 \leq 1.0$ **Then** C_0
Else

> **If** $x_4 > 0.5$ **and** $x_4 \leq 1.0$ **and** $x_1 \leq 1$ **Then** C_0
> **Else**
>
> > **If** $x_4 > 1.0$ **and** $x_4 \leq 1.5$ **and** $x_2 \leq 1$ **Then** C_0
> > **Else**
> >
> > > **If** $x_4 > 1.5$ **and** $x_4 \leq 2.0$ **and** $x_1 \leq 1$ **Then** C_0
> > > **Else** C_1

Fig. 8. Example for algorithm VC-Dimension2 in Figure 7 for continuous data with $d = 4$ and $m = 12$. Using the spared feature 4 in all rules, one divides the class labelings into 4 subproblems of m = 3. Each subproblem can then be shattered with the remaining features. For the example rule, the class labeling of S is {1, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0}.

IV. Conclusion

In this paper we try to extend the work on VC-dimension in statistical learning theory, where there is no explicit formula for the VC-dimension of a rule set. In this work, we focused on the easiest case of rule sets defined on datasets with binary features. Starting from basic rule set with a single rule consisting of a single decision condition, we give and prove lower bounds of the VC-dimension of different rule set structures. We also show that our approach can be generalized to rule sets defined on datasets with continuous features. In general, we prove that the VC-dimension of a rule set depends on the number of features and the rule set structure.

References

[1] J. Fürnkranz, “Separate-and-conquer learning,” *Artificial Intelligence Review*, vol. 13, pp. 3–54, 1999.

[2] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Meteo, CA: Morgan Kaufmann, 1993.

[3] E. Frank and I. H. Witten, “Generating accurate rule sets without global optimization,” in *Proceedings of the 15th International Conference on Machine Learning*, 1998, pp. 144–151.

[4] P. Clark and R. Boswell, “Rule induction with CN2: Some recent improvements,” in *Lecture Notes in Artificial Intelligence*, vol. 482, 1990, pp. 151–163.

[5] W. W. Cohen, “Fast effective rule induction,” in *The Twelfth International Conference on Machine Learning*, 1995, pp. 115–123.

[6] V. Vapnik, *The Nature of Statistical Learning Theory.* New York: Springer Verlag, 1995.

[7] Y. Mansour, “Pessimistic decision tree pruning based on tree size,” in *Proceedings of the 14th international conference on Machine learning*, 1997.

[8] H. U. Simon, “The vapnik-chervonenkis dimension of decision trees with bounded rank,” *Information Processing Letters*, vol. 39, no. 3, pp. 137–141, 1991.

[9] O. Maimon and L. Rokach, “Improving supervised learning by feature decomposition,” in *Proceedings of the Second International Symposium on Foundations of Information and Knowledge Systems*, 2002, pp. 178–196.

[10] O. T. Yıldız, “On the vc-dimension of univariate decision trees,” in *1st International Conference on Pattern Recognition and Methods*, 2012, pp. 205–210.