ROUGH SET REDUCTION OF ATTRIBUTES AND THEIR DOMAINS FOR NEURAL NETWORKS

JACEK JELONEK, KRZYSZTOF KRAWIEC, ROMAN SŁOWKIŃSKI
Institute of Computing Science, Poznan University of Technology, PL-60-965 Poznan, Poland

This paper presents an empirical study of the use of the rough set approach to reduction of data for a neural network classifying objects described by quantitative and qualitative attributes. Two kinds of reduction are considered: reduction of the set of attributes and reduction of the domains of attributes. Computational tests were performed with five data sets having different character, for original and two reduced representations of data. The learning time acceleration due to data reduction is up to 4.72 times. The resulting increase of misclassification error does not exceed 11.06%. These promising results let us claim that the rough set approach is a useful tool for preprocessing of data for neural networks.

Key words: neural networks, rough set, reduction, quantitative and qualitative attributes, computational experiment.

1. INTRODUCTION

While many controversies still rage over the usefulness of neural networks (NN), at least one fact is clear: most of them are very complex computationally. The time required by their learning algorithms is in most cases incomparably greater than that required by other methods used in artificial intelligence, e.g., by machine learning algorithms. Therefore it is not surprising that there is so much interest in accelerating the process of NN learning. In particular, since Rumelhart’s backpropagation has come into being (Rumelhart et al. 1986), a significant amount of effort in this branch is dedicated to reducing the learning time.

In this paper, we stress the importance of preprocessing data to eliminate their redundancy before the network learning. Assuming that the task of the learning algorithm is to teach the network to classify a set of objects (data set), described by a set of attributes (description vector), into particular classes, at least two types of redundancy may occur in the data set:

1. the redundancy of an entire attribute, and
2. the redundancy of particular values of an attribute.

The elimination of (1) as well as (2) leads to a reduction in NN input size. Consequently, the number of input nodes and weights decreases, and a smaller network usually means shorter learning time. Moreover, in some cases, when the size of the original description is very large and makes use of the more sophisticated network impossible, such reduction can allow for more complex network architectures, e.g., two hidden layers instead of one, which may increase remarkably the classification ability.

It is our goal in this paper to demonstrate experimentally that data preprocessing eliminates the above-mentioned redundancies, resulting in a shortening of network learning time without a significant decrease in its classification ability.

The central question related to data preprocessing is how to reduce the description of a set of objects. In our research, a very promising approach, rough set theory (RS), proposed by Pawlak (1982, 1991), has been used for this purpose. In section 2, we show how one can make use of this theory for the reduction mentioned earlier.

Though the most natural reduction carried out using the rough set approach concerns the set of attributes (cf. Pawlak 1991), it is worth extending the reduction to the domains of attributes. So, we distinguish in this paper the reduction of attributes (AR, Attribute Reduction) from the reduction of their domains (ADR, Attribute Domain Reduction). This
work continues our earlier research (Jeleńek et al. 1993), which gave the first encouraging result of data reduction for NN classifying descriptions of histological pictures.

In the next section, we characterize the attribute coding for the NNs. Then, in section 3, the two types of reduction of object description are presented. Section 4 describes the computational tests. The last, fifth section includes a discussion of the results.

2. ATTRIBUTE CODING FOR THE NEURAL NETWORK

Each of the description attributes can be assigned to one of the following groups:

- **Quantitative attributes.** These attributes represent some measurable properties of objects. Their values are ordered by definition. A quantitative attribute can be (after optional scaling) directly used as input to an NN.
  - Examples: length, temperature, cost.

- **Qualitative attributes.** This group includes the attributes for which the values are expressed in linguistic terms. It can be divided into two subgroups:
  - **Ordered qualitative attributes.** The values of these attributes can be ordered according to some **axis of significance**. The order of linguistic values can be represented by a sequence of increasing or decreasing numbers encoding them. Similar to quantitative attributes, they can be easily coded and then used as input data for NN.
    - Example: age = (young, medium, old); exemplary coding: young = 0.0, medium = 0.5, old = 1.0.
  - **Unordered qualitative attributes** (also called nominal). The linguistic values of these attributes cannot be ordered; in other words, it is impossible to arrange them on some axis of significance. Theoretical considerations (e.g., Cherasky and Lari-Najafi 1992) as well as practical experience show that it is extremely inconvenient to treat these attributes as ordered ones.
  - Unordered qualitative attributes need additional coding: each linguistic value is then represented by separate input to the NN. This encoding, called one-from-n (or one-in-n), creates a binary vector whose elements correspond to the network inputs. When an attribute takes a particular value, the corresponding vector element is set equal to 1, while others are set to 0.
    - Examples: mark-of-the-car, sex, color.

Thus, while networks can deal very well with ordered attributes, the nominal (unordered qualitative) attributes almost always cause an extension of the NN architecture.

3. REDUCTION OF DESCRIPTION USING THE ROUGH SET APPROACH

The approach based on the rough set theory deals with inconsistency in the data set. Inconsistency means there are at least two objects with the same description (i.e., vector of attribute values), but with distinct classification. This often occurs in naturally collected (i.e., not artificially created) data sets, which usually indicates that the description is probably insufficient to distinguish objects. Inconsistency is characterized in the rough set theory by the notion of the quality of approximation of the classification or, shortly, the quality of classification $\gamma$. It is computed on the base of the sole data set and indicates the accuracy of the classification by the given set of attributes. It is equal to 1.0 when there is no inconsistency in the data set; otherwise it is between 0 and 1. The quality of classification is an intrinsic characteristic of the particular data set; it should not be confused with the classification ability or true-ratio, which is used for learning systems evaluation.

Besides inconsistency, rough set theory deals with redundancy typical of object description. Usually it is possible to reduce the redundant description without loss of information relevant for the classification. In terms of the rough set theory, this corresponds to the reduction of the set of attributes (AR) and to the reduction of the domains of attributes (ADR).

3.1. Attribute Reduction

Reduction AR tends to find a reduct of the set of attributes that preserves the quality of classification $\gamma$. For one data set, many reducts may exist; they all have the property that removing any attribute results in decreasing $\gamma$. We mostly interested in the smallest reduct. Finding it, however, is in general an NP-complete problem, so we are satisfied with the following approximate procedure:

1. Compute the classification quality $\gamma_0$ for the original set of attributes $A$.
2. Compute the core $C$, i.e., the intersection of all reducts.\(^1\) For the core, the classification quality $\gamma_C \leq \gamma_0$, by definition. In particular, the core may be empty and then $\gamma_C = 0$.
3. Set $R := C$. For $A$ being the set of attributes repeat:
   \begin{enumerate}
   \item Compute the gain of the quality of classification $\gamma$ for every attribute $a \in A \setminus R$ appended to $R$:
     \[ \text{gain} := \gamma_{R \cup \{a\}} - \gamma_R \]
   \item Choose attribute $a$ giving the biggest\(^2\) increase of $\gamma$ and set $R := R \cup \{a\}$.
   \item If $\gamma_R = \gamma_0$ then stop, otherwise go to step (a).
   \end{enumerate}
4. The result $R$ constitutes the reduct which is possibly, but not necessarily, the smallest one.

Here is an exemplary reduction carried out in this manner:

**Example 1.** Data set *Buses*, described by eight attributes.

1. Classification quality for the original description is $\gamma_0 = 1.0$.
2. Computed core includes only one attribute: $C = \{3\}$; $\gamma_C = 0.487$.
3. $\gamma(C) < \gamma_0$, so $C$ does not constitute a reduct. Therefore, compute the gain of the classification quality for every remaining attribute appended to $R = C$:

<table>
<thead>
<tr>
<th>Attribute $a$</th>
<th>$\gamma_{R \cup {a}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.737</td>
</tr>
<tr>
<td>2</td>
<td>0.711</td>
</tr>
<tr>
<td>3</td>
<td>0.487</td>
</tr>
<tr>
<td>4</td>
<td>0.487</td>
</tr>
<tr>
<td>5</td>
<td>0.487</td>
</tr>
<tr>
<td>6</td>
<td>0.487</td>
</tr>
<tr>
<td>7</td>
<td>0.487</td>
</tr>
<tr>
<td>8</td>
<td>0.487</td>
</tr>
</tbody>
</table>

The biggest gain gives attribute no. 1; therefore, $R = \{3, 1\}$ and new $\gamma_R = 0.737$.

Next steps:

- $R = \{3, 1, 4\}$
- $\gamma_R = 0.855$
- $\gamma_R = 0.974$
- $\gamma_R = 1.000$

\(^1\)To find the intersection it is not necessary to compute a priori all the reducts (cf. Slowiński and Stefanowski et al. 1992).

\(^2\)Ties are broken by the attribute number.
Computational Intelligence

After the last step, \( y_0 = y_0 = 1.000 \). The procedure is stopped and \( R = [3, 1, 4, 2, 5] \) is possibly the smallest reducet.

The set of attributes obtained in this way constitutes the reduced representation and is then the base for building the NN with reduced architecture.

3.2. Attribute Domain Reduction

In the second kind of reduction, ADR, every value of a particular attribute (in the original description) is treated as a new binary attribute (one-from-\( n \) coding). In other words, every attribute with domain cardinality equal to \( N \) is converted first into \( N \) binary attributes. Then the process of reduction concerns all binary attributes, exactly in the way described for the case of AR.

In the following example of ADR reduction we concentrate on one attribute from the Lymphography data set (see Michalski et al. 1986 for details).

Example 2.

Attribute (nominal): changes-in-structure

Domain (set of values): 1-no, 2-grainy, 3-drop like, 4-coarse, 5-diluted, 6-reticular, 7-striped, 8-faint.

The attribute belongs to the class of unordered qualitative attributes, and, consequently, it requires eight inputs to the NN. In other words, it has to be converted into eight binary attributes: 1-no-changes-in-structure, 2-grainy-changes-in-structure, 3-drop-like-changes-in-structure, ... 8-faint-changes-in-structure.

The domain of each attribute is \( (y, n) \) and, by definition, only one of the eight can be yes at a time. In this case ADR results in reduction of binary attributes \( \{1, 3, 4, 7, 8\} \). Then one remains with three binary inputs to the (reduced) NN: 2-grainy-changes-in-structure, 5-diluted-changes-in-structure, 6-reticular-changes-in-structure.

That gives the NN input reduction by 62.5% (from eight to three attributes).

4. THE COMPUTATIONAL TESTS

To investigate the virtues and the disadvantages of both AR and ADR, we performed tests of NN learning for three data representations: using the \( (1) \) not reduced (original); \( (2) \) reduced by AR; and \( (3) \) reduced by ADR data representation. For representations \( (1) \) and \( (2) \) a binarization of the original attributes has been performed in order to define the input to the NN. The original data sets are described in section 3.2.

In some cases ADR gave no decrease of the number of inputs to the NN in comparison with AR. There are two possible reasons for it:

1. As follows from section 3, it is reasonable to perform ADR on the nominal (unordered qualitative) attributes only. The Buses data set does not include any nominal attribute.

2. The only nominal (unordered qualitative) attributes are binary ones. This is the case of the Election data set.

Therefore, in the case of the Buses as well as the Election data set, no tests for ADR have been carried out.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Election Cancer</th>
<th>Buses</th>
<th>Lymphography</th>
<th>Ouko</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objects</td>
<td>444</td>
<td>285</td>
<td>76</td>
<td>148</td>
</tr>
<tr>
<td>Classes</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Distribution</td>
<td>201,233</td>
<td>84,201</td>
<td>30,46</td>
<td>2,46,61,81</td>
</tr>
<tr>
<td>among</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>classes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attributes</td>
<td>30</td>
<td>8</td>
<td>8</td>
<td>18</td>
</tr>
<tr>
<td>Nominal</td>
<td>2</td>
<td>4</td>
<td>-</td>
<td>14</td>
</tr>
<tr>
<td>Ordered</td>
<td>28</td>
<td>4</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Inconsistency</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

4.1. Architectures and Properties of NNs Used in Tests

Since the main goal of this paper is to show the suitability of AR and ADR for reduction, quite simple and unsophisticated NNs have been used in performed tests. Here is a short characterization of them (for details, see Rumelhart et al. 1986 or other works concerning backpropagation NNs).

- Each network consists of sigmoidal neurons arranged in two or three layers. In the output layer, there are as many neurons as there are classes to be learned in the data set. The classification is represented in one-from-\( n \) code \( (n = \text{number of classes}) \).
- The simplest form of backpropagation has been used as the learning algorithm: there is neither weight momentum nor other improving technique implemented.
- All the tests have been performed for only one value of the learning speed \( a = 0.8 \). The weight updates are carried out after each example.
- A classification-oriented convergence criterion has been used. We assume that the object is properly classified when all the outputs \( o_i \) of the network output layer satisfy the condition \( |o_i - d_i| < 0.25 \), where \( d_i \) denotes desired output value and \( j \) indexes the outputs.
- When the convergence criterion has not been satisfied during 250 epochs (cycles of learning the entire learning set), the learning was stopped.

4.2. Data Sets

Five data sets have been used in performed tests. Detailed descriptions of the sets Lymphography and Breast Cancer may be found in Michalski et al. 1986. The Election data set was used in Hadjimichael and Waskiewska 1992. Buses and Ouko come from our environment (see the acknowledgments). In the Breast Cancer data set, any objects with unknown attribute values were removed from the original data set.

A detailed description of these sets is presented in Table 1, which includes number of objects (examples), number of classes (categories), distribution of categories, and number of attributes, particularly the share of the nominal and ordinal attributes in the description. The last row shows that only the Breast Cancer data set includes some inconsistency, i.e., at least two objects are described by identical values of attributes but assigned to distinct classes.
TABLE 2. Description Reduction for Particular Data Sets.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Election</th>
<th>Breast</th>
<th>Busses</th>
<th>Lymphography</th>
<th>Onko</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>NN Inputs</td>
<td>32</td>
<td>14</td>
<td>8</td>
<td>35</td>
</tr>
<tr>
<td>Attribute Reduction</td>
<td>NN Inputs</td>
<td>5</td>
<td>13</td>
<td>5</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>Reduction</td>
<td>84.3%</td>
<td>7.1%</td>
<td>37.5%</td>
<td>48.5%</td>
</tr>
<tr>
<td>Attribute</td>
<td>NN Inputs</td>
<td>—</td>
<td>9</td>
<td>—</td>
<td>7</td>
</tr>
<tr>
<td>Domain Reduction</td>
<td>Reduction</td>
<td>—</td>
<td>35.7%</td>
<td>—</td>
<td>80.0%</td>
</tr>
</tbody>
</table>

4.3. Test Results

Table 2 shows the description reduction carried out through AR and ADR. The first row includes the number of NN inputs required, when using the original, not reduced description. The following rows include the size of the resulting description and the percentage of representation reduction achieved for both AR and ADR. The empty spaces for ADR show that no test was performed because of the lack of the nominal attributes (Busses) or because AR gave the same reduced description vector (Election).

The architectures of NNs used for particular data sets before and after the two kinds of reduction as well as their classification performances are characterized in Table 3. The net architecture is characterized by three numbers separated by dashes, which define the number of neurons in the consecutive layers. As mentioned earlier, due to the lack of formal methods for estimating the number of neurons in the hidden layer(s), it has been chosen heuristically, taking into account the input size, number of classes, and size of the data set (number of objects).

Table 3 also includes the number of weights for each tested network; this factor reflects the complexity of a particular network. The learning time (in minutes) is also given, and the misclassification errors are presented. For both AR and ADR, the increase or decrease in error is also shown (in comparison to the original).

The computations were carried out on a 486DX/33MHz personal computer; for this purpose, backpropagation algorithm was implemented in the C++ language. The misclassification error has been computed on the base of a tenfold cross-validation technique (for details, see Weiss and Kropoules 1989). The reduction has been carried out by means of RoughDAS system (Słowiński and Stefanowski 1992).

In the final comparison, shown in Table 4, learning time acceleration and performance index, both computed in comparison to the original networks, are presented for AR as well as ADR. Learning time acceleration is computed as a quotient of the original and reduced network learning time. The performance index is equal to the true ratio (100% minus misclassification error) of the reduced network divided by the true ratio achieved by the original network.

5. DISCUSSION

1. Rough set–based reduction of data representation appears to cooperate well with backpropagation-learned neural networks. Computational experiments show that, using described methods, it is possible to reduce significantly (even several times) the learning time, keeping the misclassification error approximately stable.

2. Small variations of the misclassification error for AR and ADR in comparison with the original data set are not uniformly positive or negative. In the case of AR for Busses and ADR for Lymphography there has been an observed improvement (over 5% and 2%,
respectively; see Table 3), but in other cases a deterioration. The possible reasons for this ambiguity are the following:

- **AR and ADR decreases the dimension of the description space (or the input space, when regarding NNs).** The network built for reduced representation has a different (smaller) architecture and weight number; consequently, the shape of the error function over the weight space changes as well. As a result, the freedom degree of the system is reduced and the risk of the local minimum trap may increase.

- **The network architecture alone can influence the results.** When the number of hidden-layer neurons is too small, the learning algorithm may be unable to teach the network to classify properly a particular data set. However, due to the distributed nature of the data processing in NNs, it is rare that removing or adding a few neurons to the hidden layer completely changes its classification ability after learning. Usually this ability changes only slightly, and the bigger the hidden layer, the smaller the influence. Thus it is better to equip a network with a too large hidden layer(s) than with a too small one(s), although there is no method for estimating the "sufficient" size.

3. **The ADR performs better than AR as far as the NN input size reduction is concerned.** Consequently, it gives smaller networks and therefore reduces the learning time. Moreover, in terms of classification ability, ADR is better than AR (see Table 3).

ACKNOWLEDGMENTS

The authors thank Jerzy Stefanowski from the Institute of Computing Science, Technical University of Poznań, for help in preparing data sets for the experiments and for many valuable comments on the experimental results. The **Onko** data set was collected by Jozef Szymański from the Department of Pathology, University School of Medicine in Poznań. The **Breast Cancer and Lymphography** data sets were received from the Machine Learning Data Base at the University of California, Irvine (see Murphy and Aha 1991). These domains came originally from the University Medical Center, Institute of Oncology, Lublińna. Thanks go to M. Zwitter and M. Soklic for providing the data. The **Bussers** data set was collected by Jacek Zaki from the Mechanical Department, Technical University of Poznań. The **Election** data set was considered in Hadimichael and Wasilewska (1992). The authors thank Dr. Taber and Dr. Churchman from the Department of Political Science, SUNY, at Stony Brook, for providing the data. This research was supported by Grant KBN No. 8 5503 016 06.

REFERENCES

