An Efficient and Effective Case Classification Method Based On Slicing

Omar A. A. Shiba, Md. Nasir Sulaiman, Ali Mamat and Fatimah Ahmad

Faculty of Computer Science and Information Technology
University Putra Malaysia, 43400 UPM Serdang, Selangor
E-mail: abumoad99@hotmail.com

Abstract

One of the most important tasks that we have to face in real world applications is the task of classifying particular situations and/or events as belonging to a certain class. In order to solve the classification problem, accurate classifier systems or models must be built. Several computational intelligence methodologies have been applied to construct such a classifier from particular cases or data. This paper introduces a new classification method based on slicing techniques that was proposed for procedural programming languages. The paper also discusses two of common classification algorithms that are used either in data mining or in general AI. The algorithms are: Induction of Decision Tree Algorithm (ID3) and Base Learning Algorithm (C4.5). The paper also studies the comparison between the proposed method and the two selected classification algorithms using several domains.

Keywords: - data mining, classification problem, classification algorithms, slicing techniques, case slicing, ID3, C4.5

1. Introduction

Classification is the most important task in machine learning. In classification, a classifier is built from a set of training examples with class labels. A key performance measure of the classifier is its predictive accuracy on the training and testing examples [1]. The classification problem has been studied extensively by the database and Artificial Intelligence communities. The problem of classification is defined as follows: The input data are referred to as the training set, which contains a plurality of records, each of which contains multiple attributes or features. Each example in the training set is tagged with a class label. The training set is used in order to build a model of the classification attribute based upon the other attributes. This model is used in order to predict the value of the class label for the test set. Some well-known techniques for classification include the following: ID3 and C4.5 Algorithm [2, 3]. This paper introduces new classification method based on slicing technique. Slicing technique was proposed for procedural programming languages [4]. Slicing is a method used by experienced computer programmers for restricting, the behaviour of program to some specified subset of interest.

The remainder of the paper is organized as follows: Section 2 presents brief description of some related work; case classification based on slicing technique is described in section 3; the experimental results and the conclusion are presented in sections 4 and 5 respectively.
2. Related Work

In this section a brief description of two of the common classification algorithms that are related to our work will be presented.

2.1 C4.5 Algorithm

C4.5 is an extension to the decision-tree learning algorithm ID3 [2, 3]. The algorithm consists of the following steps:

1. Build the decision tree form the training set (conventional ID3).
2. Convert the resulting tree into an equivalent set of rules. The number of rules is equivalent to the number of possible paths from the root to a leaf node.
3. Prune each rule by removing any preconditions that result in improving its accuracy, according to a validation set.
4. Sort the pruned rules in descending order according to their accuracy, and consider them in this sequence when classifying subsequent instances.

2.2 Induction of Decision Tree Algorithm (ID3)

ID3 is an algorithm introduced by Quinlan for inducing Classification Model, also called Induction of Decision Tree, from data. We are given a set of records. Each record has the same structure, consisting of a number of attribute / value pairs. One of these attributes represents the category of the record. The problem is to determine a decision tree that on the basis of answers to questions about the non-category attributes predicts correctly the value of the category attribute. Usually the category attribute takes only the values \{true, false\}, or \{success, failure\}, or something equivalent. In any case, one of its values will mean failure [2, 3]. The basic ideas behind ID3 are that:

In the decision tree each node corresponds to a non-categorical attribute and each arc to a possible value of that attribute. A leaf of the tree specifies the expected value of the categorical attribute for the records described by the path from the root to that leaf. (This defines what is a Decision Tree.)

In the decision tree at each node should be associated the non-categorical attribute which is most informative among the attributes not yet considered in the path from the root. (This establishes what is a “Good” decision tree.)

Entropy is used to measure how informative is a node. (This defines what we mean by “Good”. By the way, this notion was introduced by Claude Shannon in Information Theory) [5].

3. Case Classification Based On Slicing

In this section the proposed classification approach and some related terms are discussed.

3.1 Program Slicing Technique

Slicing is a method used by experienced computer programmers for restricting, the behaviour of program to some specified subset of interest [4, 6]. A slice is constructed by deleting those parts of the program that are irrelevant to the values stored in the chosen set of variables at the chosen point. The point of interest is usually identified by annotating the program with line numbers, which identify each primitive statement and each branch node. Program slicing is useful for program understanding, maintenance, debugging, testing, differencing, specialization, reuse, optimization, parallelization, and anomaly detection [7]. Program slicing has been widely studied in the context of imperative programs. Several slicing algorithms for imperative languages have been developed
An Efficient and Effective Case Classification Method Based On Slicing

Slicing of programs is performed with respect to some criterion. Weiser proposes as a criterion the number of a command line and a subset of program variables [4]. According to this criterion, a program is analyzed and its commands are checked for their relevance to command line and those variables in V. However, other authors have defined different criterion [9]. Program slicing can be summarized as following:

**Program Slice**
≡ the statements (and predicates) that might affect the value of a set of variables at a particular statement.

A slice is taken with respect to a set of variables at a particular statement, the slicing criterion.
A slice and the actual program behavior are identical.
Not just the final value, but all intermediate steps

**Executable (Slice)**
≡ a slice that can be compiled and executed.

**Closure (Slice)**
≡ an informational presentation of a slice that might lack semantics.

Basic Types:
- Static vs. Dynamic
- Type of feedback: executable, closure
- Approach: graph reachability, dataflow equations using control flow.

### 3.2 Extending Program Slicing To Case Slicing

The case slicing classifier described in this paper address the problem of classification. The case slicing classifier is extended of program slicing technique. When we slice a case we are interested in automatically obtaining that portion ‘features’ of the case responsible for specific parts of the solution of the case at hand.

A case slicing: is a process for automatically obtaining subparts (features) of a case with a collective meaning.

A slicing criterion: denotes the conditions of the slice computation, with respect to which and for which case a slice is required.

Sliced case: contains all features that could have direct relations with the features of interest at new case.

### 3.3 The Case Slicing Technique

Conceptually, the proposed method is a variation of the Nearest Neighbor Algorithms [10] and is called Case Slicing Technique (CST). It compares new cases to the training cases in the data file. It computes the similarity between the new cases and training cases to classify the new cases. The proposed method is a classification technique based on slicing. Slice case means we are interested in automatically obtaining that portion ‘features’ of the case responsible for specific parts of the solution of the case at hand. By slicing the case with respect to important features we can obtain new case with a small number of features or with only important features.

The proposed approach consists of a database with three calculation modules as follows:

- **Features Weighting Module**

  This module is used to measure the importance of each attribute in classification. The weight of each attribute has been calculated to classify the new case by using simple conditional probabilities. High weight values were assigning to features that are highly correlated with the given class using equation (1).

$$w_{ia} = P(C|ia) = \frac{\text{instances containing } ia \land \text{class } C}{\text{instances containing } ia}$$

Where the weight for feature a for a class c is the conditional probability that a case is a member of c given the value to a.
Discretization Computing Module

Discretization as used in this paper, and in the machine learning literature in general, is a process of transforming a continuous attribute values into a finite number of intervals and associating with each interval a discrete, numerical value. The usual approach for learning tasks that use mixed-mode (continuous and discrete) data is to perform discretization prior to the learning process [11, 12, 13, 14].

The discretization process first finds the number of discrete intervals, and then the width, or the boundaries for the intervals, given the range of values of a continuous attribute. Most often the user must specify the number of intervals, or provide some heuristic rule to be used [15].

A variety of discretization methods have been developed in recent years. Some models that have used the Value Difference Metrics (VDM) or variants of it [16, 17, 18] have discretized continuous attributes into a somewhat arbitrary number of discrete ranges, and then treated these values as nominal (discrete unordered) values.

When using slicing approach, continuous values are discretized into s equal-width intervals (though the continuous values are also retained for later use), where s is an integer supplied by the user. Unfortunately, there is currently little guidance on what value of s to use. Current research is examining more sophisticated techniques for determining good values of s, such as cross-validation, or other statistical methods [19]. The width \( w_a \) of a discretized interval for attribute \( a \) is given by equation (2).

\[
\frac{\max_a - \min_a}{s}
\]

(2)

where \( \max_a \) and \( \min_a \) are the maximum and minimum value, respectively, occurring in the training set for attribute \( a \).

The discretized value \( v \) of a continuous value \( x \) for attribute \( a \) is an integer from 1 to \( s \), and is given by equation (3).

\[
v = \text{dis}_a(x) = \begin{cases} 
\frac{(x - \min_a)}{w_a} & \text{if attribute } a \text{ is continuous} \\
\frac{x}{w_0} & \text{if attribute } a \text{ is discrete}
\end{cases}
\]

(3)

Distance Computation Module

There are many learning systems that store some or all available training examples during learning. During generalization, a new input vector is presented to the system for classification and a distance function is used to determine how far each stored instance is from the new input vector. The stored instance or instances which are closest to the new vector are used to classify it. A variety of distance functions are available for such uses, including the Minkowsky [20], Mahalanobis [21], Canberra, Chebychev, Quadratic, Correlation, and Chi-square distance metrics [22, 23], the Context-Similarity measure [24], the Contrast Model [25], hyperrectangle distance functions [26, 27] and others.

Although there have been many distance functions proposed, by far the most commonly used is the Euclidean distance function, which is defined in equation (4).

\[
E(x, y) = \sqrt{\sum_{a=1}^{m} (x_a - y_a)^2}
\]

(4)

Where \( x \) and \( y \) are two input vectors (one typically being from a stored instance, and the other an input vector to be classified) and \( m \) is the number of input variables.
(attributes) in the application. The square root is often not computed in practice, because the closest instance(s) will still be the closest, regardless of whether the square root is taken.

An alternative function, the City-block or Manhattan distance function, requires less computation and is defined in equation (5).

\[
M(x, y) = \sum_{a=1}^{n} |x_a - y_a| 
\]

(5)

The Euclidean and Manhattan distance functions are equivalent to the Minkowskian r-distance function [20] with \( r = 2 \) and \( 1 \), respectively.

3.3.1 Slicing Technique

The objective of slicing technique is to optimize the similarity matching to achieve best classification results. The proposed approach is adapting slicing techniques that have been used in programming languages, to slice the cases by removing subset of features which are irrelevant to case label with respect to the selected slicing criterion. Case classification algorithm based slicing is shown in Figure 1.

3.3.2 A Formal Description Of Case Slicing Technique

In this section we will give a formal description of Case Slicing Techniques in a basic version allowing for a detailed investigation of the approach.

Let

\[
S = \{C_1, C_2, C_3 \ldots C_n\} \text{ set of cases in Case Base} \\
\forall S \exists C_i \quad S \neq \emptyset
\]

\( C_i = \{f_1, f_2, f_3 \ldots f_n\} \) where \( n \) is the number of features in \( C_i \)

\( \lambda = \{\{C_s|C_s \text{ is a set of sliced cases}\}\} \) OR

\( \lambda = \{\text{all cases that contains one or more important feature(s)}\} \)

\( I = \{if_1, if_2, \ldots, if_n\} \) where \( n \) is the number of important features in \( I \)

\( I \subseteq C_i \subseteq S \)

\( I \subseteq C_s \subseteq \lambda \)

4. Experimental Results

In this section the results of several practical experiments are presented to examine the performance of the proposed approach and the performance of the selected classification algorithms on real world problems.

4.1 Selected Datasets

In this paper eight real-world datasets have been used, which are widely used in the machine-learning field for evaluation of case slicing technique. The eight datasets: Cleveland Heart Disease (CLEV), Breast Cancer (BCO), German Credit Card (GERM), Hepatitis Domain (HEPA), Australian Credit Card Approval (AUS), Iris Plants Database (IRIS), United States Congressional Voting Records Database (VOTING) and Credit Card Application (CRX) were chosen from the UCI: Machine Learning Repositories and Domain Theories [28]. Table 1 presents the main characteristics of these datasets, where B, C and D in the table means Boolean, continuous and discrete attributes respectively.
Algorithm: Algorithm for Case Classification
Input: User’s Input Problem Specification
Output: Classified Case
BEGIN
While True do

**Step 1** Discretize Continuous Values
Let \( x \) be the input value for attribute \( a \) of case \( i \)
\[
\begin{align*}
    v &= \text{disc}_a(x) \\
    w_a &= \frac{\text{abs}(\max_a - \min_a)}{s}
\end{align*}
\]
{The width of a discretized interval for attribute \( a \)}
{Where \( \max \) and \( \min \) are the maximum and minimum value, respectively, which are occurring in
the training set for attribute \( a \).} {the discretized value \( v \) of a continuous value \( x \) for attribute \( a \) is
an integer from 1 to \( s \) and \( s \) determine by the user.}
If \( a \) is continuous then
\[
    v = \text{disc}_a(x) = \left\lfloor x - \min_a \right\rfloor / w_a
\]
Else
\[
    v = \text{disc}_a(x) = x
\]
Endif

**Step 2** Assign Weights to features {using conditional probability}
Let \( c \) be the output class of case \( i \)
\[
\begin{align*}
    N_{a,v,c} &= N_{a,v,c} + 1 \quad \{\text{# of value } v \text{ of attribute } a \text{ with output class } c\} \\
    N_{a,v} &= N_{a,v} + 1 \quad \{\text{# of value } v \text{ of attribute } a\}
\end{align*}
\]
For each value \( v \) (of attribute \( a \))
For each class \( c \)
If \( N_{a,v} = 0 \)
\[
    p_{a,v,c} = 0
\]
Else
\[
    p_{a,v,c} = \frac{N_{a,v,c}}{N_{a,v}}
\]
ENDFOR
ENDFOR

**Step 3** Case Slicing
For each case \( i \) in \( T \) {\( T \)-Training set}
Slice case \( i \) w.r.t. selected criterion
{removing irrelevant attributes depending on attribute weights}
Endfor

**Step 4** Distance Calculations
For each two cases \( x \) and \( y \)
{One typically being from stored cases, and other the input case to be classified}
Let \( m \) the number of attributes in the case
For \( a = 1 \) to \( m \) do
\[
    \text{distance}(x, y) = \text{abs}(x_a - y_a)
\]
Distance_Summation {the cumulative distance of all the \( m \) attributes in
the case \( i \)}
Endfor

**Step 5** Closer Case Searching
While not done do
Find matching between cases to get closer case with less distance
Enddo

Figure 1. Case classification algorithm based slicing
Table 1. Characteristics of the selected datasets

<table>
<thead>
<tr>
<th>Datasets</th>
<th>No. Of Data</th>
<th>Type &amp; No. Of Attributes</th>
<th>No. Of Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLEV</td>
<td>303</td>
<td>6C, 9D (15)</td>
<td>2</td>
</tr>
<tr>
<td>BECO</td>
<td>699</td>
<td>13B, 6C (19)</td>
<td>2</td>
</tr>
<tr>
<td>GERM</td>
<td>1000</td>
<td>16B (16)</td>
<td>2</td>
</tr>
<tr>
<td>HEPA</td>
<td>155</td>
<td>13B, 6C (19)</td>
<td>2</td>
</tr>
<tr>
<td>AUS</td>
<td>690</td>
<td>6C, 9D (15)</td>
<td>2</td>
</tr>
<tr>
<td>IRIS</td>
<td>150</td>
<td>4C (4)</td>
<td>3</td>
</tr>
<tr>
<td>VOTING</td>
<td>435</td>
<td>16B (16)</td>
<td>2</td>
</tr>
<tr>
<td>CRX</td>
<td>690</td>
<td>6C, 9D (15)</td>
<td>2</td>
</tr>
</tbody>
</table>

4.2 Empirical Results

We evaluated the performance of the case slicing technique by comparing it against the ID3 and C4.5 classifiers on a variety of datasets. The datasets we have selected are very good choice to test and evaluate the slicing technique because the datasets are from different domains and there is a good mixture of continues, discrete and Boolean features. In all the experiments reported here we used the evaluation technique 10-fold cross-validation, which consists of randomly dividing the data into 10 equally sized subgroups and performing ten different experiments. We separated one group along with their original labels as the validation set; another group was considered as the starting training set; the remainder of the data were considered the test set. Each experiment consists of ten runs of the procedure described above, and the overall average are the results reported here. The criterion of choosing the best classification approach is based on the highest percentage of classification. The results, given in Table 2, list the classification accuracies achieved by each approach for each of the datasets, and Figure 2 shows the difference in classification accuracy.

Table 2. The classification accuracy achieved by the different classification algorithms.

<table>
<thead>
<tr>
<th>Methods</th>
<th>C4.5</th>
<th>ID3</th>
<th>CST</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLEV</td>
<td>77.20</td>
<td>71.20</td>
<td>96.00</td>
</tr>
<tr>
<td>BCO</td>
<td>96.00</td>
<td>94.30</td>
<td>99.30</td>
</tr>
<tr>
<td>GERM</td>
<td>98.50</td>
<td>63.00</td>
<td>98.00</td>
</tr>
<tr>
<td>HEPA</td>
<td>80.80</td>
<td>67.74</td>
<td>97.00</td>
</tr>
<tr>
<td>AUS</td>
<td>84.50</td>
<td>78.26</td>
<td>99.30</td>
</tr>
<tr>
<td>IRIS</td>
<td>94.67</td>
<td>96.67</td>
<td>99.30</td>
</tr>
<tr>
<td>VOTING</td>
<td>95.63</td>
<td>93.10</td>
<td>97.00</td>
</tr>
<tr>
<td>CRX</td>
<td>85.80</td>
<td>79.71</td>
<td>97.80</td>
</tr>
</tbody>
</table>
5. Conclusion

This paper has presented and discussed the Case Slicing Technique (CST) as a new approach based slicing to improve classification task in data mining. CST was supported with experiments on eight datasets. The experiments show that using the CST indeed improves the accuracy of classification. The paper also gave brief description of a number of common classification algorithms that are used either in data mining or in general AI. The paper has presented a comparison between proposed method and other selected classification algorithms using several domains. The proposed technique has possessed a competitive result. It gave very high percentage of classification accuracy.

References


An Efficient and Effective Case Classification Method Based On Slicing


