Journal of ICT, 2 (2), pp: 105-117

AN EXPERIMENTAL EVALUATION OF CASE SLICING AS A NEW CLASSIFICATION TECHNIQUE

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

Faculty of Computer Science and Information Technology,
University Putra Malaysia
43400 UPM Serdang, Selangor, Malaysia

*Email- *abumoad99@hotmail.com, **{nasir, fatimah, ali}@fsktm.upm.edu.my*

ABSTRACT

Several classification techniques are designed to discover such classifications when the classifications are unknown. The techniques are tested and evaluated, however, by matching the classifications they recover against expected classifications. Several such techniques may be compared by experimentally evaluating their performance on the same datasets. The goal of this paper is to evaluate the case slicing technique as a new classification technique. The paper achieves this goal in three steps: Firstly, it introduces the case slicing technique as a new approach. Secondly, the paper presents applications of this technique on several datasets. Lastly, it compares the proposed approach with other selected approaches such as the K-Nearest Neighbour (K-NN), Base Learning Algorithm (C4.5) and Naïve Bayes classifier (NB) in solving the classification problems. The results obtained show that the proposed approach is a promising method in solving decision-making problem.

Keywords: data mining, case-slicing technique, K-Nearest Neighbour, Base Learning Algorithm, Naïve Bayes, classification accuracy.

1.0 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 (Ling and Zhang, 2002). The problem of classification has been studied extensively by the Database and Artificial Intelligence communities. The problem of classification is defined as follows: The input data is 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: *K-Nearest Neighbour Techniques* (Aggarwal and Philip, 1998), *C4.5 Algorithm* (Quinlan, 1986; 1993) and *Naïve Bayes classifier* (Mitchell, 1997). This paper introduces a new classification method based on slicing techniques. The Slicing techniques were originally used in the area of software development (Weiser, 1984). Slicing is a method used by experienced computer programmers for restricting the behaviour of a program to some specified subset of interest. Several slicing algorithms for imperative languages have been developed. Slicing of programs is performed with respect to some criterion; Weiser proposed as a criterion the number i of a command line and a subset V of program variables. According to this criterion, a program is analyzed and its commands are checked for their relevance to command line i and those variables in V. However, other authors have defined different criterion (Vasconcelos, 2000; Kamkar, 1995; Tip, 1995). The remainder of the paper is organized as follows: Section 2 presents a brief description of some selected classification algorithms; case slicing technique is described in section 3; the experimental results are discussed in section 4. The conclusion is presented in section 5.

2.0 SELECTED CLASSIFICATION ALGORITHMS

In this section, a brief description of the selected classification methods is given. More information about *K-NN* can be found in Xiaoli (1999); Wettschereck and Aha (1995), and about *C4.5* see Quinlan (1986) (1993) and for *NB* see Mitchell (1997).

2.1 K-Nearest Neighbour (K-NN)

The basic idea of the K-Nearest Neighbour algorithm *(K-NN)* is to compare every attribute of every case in the set of similar cases with every corresponding attribute of the input case. A numeric function is used to decide the value of comparison. The *K-NN* algorithm then selects a case, which has the highest comparison value and retrieves it (Xiaoli, 1999).

K-NN assumes each case $X = \{x1, x2, \ldots, xn, x_c\}$ is defined by a set of n (numeric or symbolic) features, where x_c is x's class value. Given a query q and case library L, *k-NN* retrieves the set k of q's k most similar (i.e., least distant) cases in L and predicts their weighted majority class as the class of q (Wettschereck and Aha, 1995). Distance in *K-NN* is defined as in equation (1).

$$\text{distance}(x,q) = \sqrt{\sum_{f=1}^{n} w_f * difference(xf, qf)^2} \quad , \tag{1}$$

where w_f is the parameterized weight value assigned to feature f as in equation (2).

$$w_f = P(C|ia). \tag{2}$$

That is, 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 where $P(C \mid i_a)$ is defined in equation (3); and the difference between x and q can be calculated as in equation (4).

$$P(C|ia) = \frac{|instances \ \ containing \ \ i_a \wedge class = C|}{|instances \ \ containing \ \ i_a|} \quad , \tag{3}$$

$$difference(x_f, q_f) = \begin{cases} |x_f - q_f| & \text{if feature f is numeric} \\ 0 & \text{if feature f is symbolic} \ \ \& \ x_f = q_f \\ 1 & \text{otherwise} \end{cases} \quad . \tag{4}$$

In equation (2), *K-NN* assigns equal weights to all features (i.e. $\forall f \{wf = 1\}$).

2.2 The Base Learning Algorithm C4.5

C4.5 is an extension to the Decision-Tree Learning Algorithm ID3 (Quinlan, 1986; 1993). The algorithm consists of the following steps:

1. Build the decision tree from 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.

2.3 Naïve Bayes (NB)

The estimates of the probability masses are used as input for a Naïve Bayes classifier. This classifier simply computes the conditional probabilities of the different classes given the values of attributes and then selects the class with the highest conditional probability. If an instance is described with n attributes a_i $(i=1,\ldots,n)$, then the class of that instance is classified to a class v from a set of possible classes V according to a Maximum A Priori criterion (MAP). The Naive Bayes classifier can be defined as:

$$v = \arg\max_{v_j \in V} p(v) \prod_{i=1}^{n} p(a_i \mid v_j) \quad . \qquad (5)$$

The conditional probabilities in the above formula are obtained from the estimates of the probability mass function using the training data. This Bayes classifier minimizes the probability of classification error under the assumption that the sequence of points is independent (Mitchell, 1997).

3.0 THE PROPOSED APPROACH

Conceptually, the proposed method is a variation of the nearest neighbour algorithms and is called Case Slicing Technique (CST). It compares new cases to the training cases in the data file. Likewise, it computes the similarity between the new cases and training cases to classify the new cases. For our purpose in this paper, we use the similarity function based on the distance

concept. The most used similarity function is the Nearest Neighbor algorithm, which computes the similarity between two cases using a global similarity measure (Aha, 1998).

The proposed method is a classification technique based on slicing. Slice case means we are interested in automatically obtaining that portion of the 'features' 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 a new case with a small number of a feature or with only important features. The proposed approach consists of a database with three calculation modules as follows:

3.1 Features Weighting Module

This module is used to measure the importance of each attribute in the classification. The weight of each attribute has been calculated to classify the new case by using simple conditional probabilities. High weight values are assigned to features that are highly correlated with the given class, using equations (2), and (3) in section 2.1.

3.2 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 value 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 (Catlett, 1991; Dougherty et al., 1995; Fayyad and Irani, 1992; Pfahringer, 1995).

The discretization process 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 (Ching et al., 1995).

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 (Cost and Salzberg, 1993; Rachlin et al., 1994; These Mohri and Tanaka, 1994) have discretized continuous attributes into an arbitrary number of discrete ranges, and then treated these values as nominal (discrete unordered) values.

When using the 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 (Wilson and Martinez, 1996). The width w_a of a discretized interval for attribute a is given by equation (6).

$$w_a = \frac{|max_a - min_a|}{s} \tag{6}$$

where max_a and min_a are the maximum and minimum values occurring in the data 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:

$$v = disc_a(x) = \begin{cases} \dfrac{\lceil (x - \min_a) \rceil}{w_a} & \text{if attribute a is continuous} \\ x & \text{if attribute a is discrete} \end{cases} . \tag{7}$$

3.3 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 (Randall and Tony, 1997), Mahalanobis (Nadler and Eric, 1993), Camberra, Chebychev, Quadratic, Correlation, and Chi-square distance metrics (Michalski et al., 1981; Edwin, 1974) , the Context-Similarity measure (Biberman, 1994), the Contrast Model (Tversky, 1977), hyperrectangle distance functions (Salzberg, 1991; Domingos, 1995) and others.

Although there are many distance functions that are being proposed, by far the most commonly used is the Euclidean distance function, which is defined as:

$$E(x,y)=\sqrt{\sum_{a=1}^{m}(x_a,y_a)^r} \tag{8}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ 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 (9).

$$M(x,y)=\sum_{a=1}^{m}|x_a-y_a|^r \tag{9}$$

The Euclidean and Manhattan distance functions are equivalent to the Minkowskian r-distance function (Randall and Tony, 1997) with $r = 2$ and 1, respectively.

3.4 Slicing Technique

The objective of this technique is to optimize the similarity matching in order to achieve the best classification results. The proposed approach is adapting this technique, which has been used in programming languages, to reduce the number of features in a case by selecting a subset of features using a selected slicing criterion. The slicing criterion that has been used in this technique is the <important features> in each case, which has high weights. The Case classification algorithm is shown in Fig.1.

```
Algorithm: Algorithm for Case Classification
Input:          User's Input Problem Specification
Output:         Classified Case
Begin
While True do
        Discrete_Continuous_Values()
        Assign_Weights_Cases()
        Slicing_Cases_w.r.t. Slicing_Criterion()
        Calculate_Distance()
        Closer_ Case_Searching()
        Return_Classified_Case()
Enddo
End.
```

Fig. 1 Case classification algorithm

Below is a formal description of a basic Case Slicing technique.

Let

$S = \{C_1, C_2, C_3, \ldots, C_n\}$ *set of cases in Case Base*

$\forall S \ \exists C_i \ \ S \neq \emptyset$,

where

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

$\lambda = [\{C_s \mid C_s$ *is a set of sliced cases*$\}]$ OR

$\lambda = \{$*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 , and*

$I \subseteq C_i \subseteq S$

$I \subseteq C_s \subseteq \lambda$

4.0 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, five real-world datasets have been used (which are widely used in the machine-learning field) to evaluate the case slicing technique. The five datasets: Breast Cancer (BCO), German Credit Card (GERM), Hepatitis Domain (HEPA), Australian Credit Card Approval (AUS) and Cleveland Heart

Disease (CLEV) are chosen from the *University of California Irvine (UCI): "Machine Learning Repositories and Domain Theories"* (Murphy, 1996). Table 1 presents the main characteristics of these datasets, where *B*, *C* and *D* in the table means Boolean, continuous and discrete attributes respectively.

Table 1: Characteristics of the selected datasets

Datasets	No. of Data	Type & No. of Attributes	No. of Classes
BCO	699	13B, 6C (19)	2
GERM	2000	7C, 13D, 1B (21)	2
HEPA	155	13B, 6C (19)	2
AUS	690	6C, 9D (15)	2
CLEV	303	9D, 6C (15)	2

4.2 Empirical Results

This study has evaluated the performance of the case slicing technique by comparing it with *Naive Bayes*, *Base Learning Algorithm* and *K-Nearest-Neighbour* classifiers on datasets. The certain selected datasets are a very good choice to test and evaluate the slicing technique since they are from different domains and contain a good mixture of continuous, discrete and Boolean features. In all the experiments reported here, we use 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 separate one group along with their original labels as the validation set; another group is 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 is the results reported here. The criterion of choosing the best classification approach is based on the highest percentage of classification. Fig. 2 shows how the procedure of the 10- cross validation works. The results, given in Table 2, list the classification accuracies achieved by each approach for each dataset.

- Split the data to k sets of approximately equal size (and class distribution, if stratified)
- For i=1 to k:
  –Use i^{th} subset for testing and remaining (k-1) subsets for training
- Compute average accuracy
- K-fold CV can be repeated several times, say, 100 times

Fig. 2: K-cross validation process

In Table 1, we can see that, C4.5 gives a good classification accuracy on GERM dataset. NB gives a good result on BCO dataset. K-NN gives a good classification result on both BCO and HEPA datasets, where the proposed approach gives a very good accuracy on most selected datasets compared with the other approaches.

Table 2: The classification accuracy achieved by different classification algorithms

Methods / Datasets	C4.5	NB	K-NN	CST
BCO	74.70	96.40	97.10	99.30
GERM	98.50	70.30	69.40	98.00
HEPA	80.80	86.30	92.90	97.00
AUS	84.50	84.90	81.90	99.30
CLEV	77.20	83.40	71.20	96.00

5.0 CONCLUSION

This paper has presented and discussed the Case Slicing Technique (CST) as a new approach based on slicing to improve the classification task in data mining. CST is supported with experiments on five datasets. The experiments have shown that using the CST indeed improves the accuracy of classification. The paper also gives a brief description of a number of common classification algorithms used either in data mining or in general, artificial intelligence. The paper presents a comparison between the proposed method and other selected classification algorithms in several domains. The proposed technique has shown a competitive result and very high percentage of classification accuracy.

REFERENCES

Aggrawal, C.C., & Philip, S.Y. (1998). Data mining techniques for associations, clustering and classification. In *IBM Lecture Notes in Computer Science.* T. J. Watson Research Center, Yorktown Heights: Springer-Verlag Heidelberg.

Aha, D.W. (1998). *Feature weighting for lazy learning algorithms.* (Technical Report AIC-98-003). Washington, DC: Naval Research, Laboratory, Navy Center for Applied Research in Artificial Intelligence.

Biberman, Y. (1994). A context similarity measure. In *Proceedings of the European Conference on Machine Learning (ECML-94).* (pp. 49-63) Catalina, Italy: Springer Verlag.

Catlett, J. (1991). On changing continuous attributes into ordered discrete attributes. In *Proceedings of the European Working Session on Learning,* (pp. 164-178) Porto, Portugal.

Ching, J.Y., Wong, A.K.C. & Chan, K.C.C. (1995). Class-dependent discretization for inductive learning from continuous and mixed mode data. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(7),* 641-651.

Cost, S., & Salzberg, S. (1993). A weighted nearest neighbor algorithm for learning with symbolic features.. *Machine Learning, 10,* 57-78.

Domingos, P. (1995). Rule induction and instance-based learning: A unified approach. In *Proceeding of the International Joint Conference on Artificial Intelligence (IJCAI-95),* (pp. 1226-1232) Montreal.

Dougherty, J., Kohavi, R., & Sahami, M. (1995). Supervised and unsupervised discretization of continuous feature. In *Proc. of the 12th International Conference on Machine Learning,* (pp. 194-202) Melbourne, Australia.

Edwin, D. (1974). Recent progress in distance and similarity measures in pattern recognition. *Second International Joint Conference on Pattern Recognition,* (pp. 534-539) Copenhagen, Denmark.

Fayyad, U.M., & Irani, K.B. (1992). On the handling of continuous-valued attributes in decision three generation. *Machine Learning, 8,* 87-102.

Kamkar, M., (1995). An overview and comparative classification of program slicing techniques. *Journal of System Software, 31,* 197-214.

Ling, X.C., & Zhang, H. (2002). Toward bayesian classifiers with accurate probabilities. In *Proceedings of the Sixth Pacific-Asia Conference on KDD,* (pp. 123-134), Taipei, Taiwan : Springer.

Michalski, R.S., Robert, E.S., & Edwin, D. (1981). A recent advance in data analysis: Clustering objects into classes characterized by conjunctive concepts. In Laveen N. Kanal and Azriel Rosenfeld (Eds.). *Progress in pattern recognition, 1*,33-56, New York: North-Holland.

Mitchell, T. M. (1997). *Machine learning.* New York:McGraw-Hill,.

Mohri, T., & Tanaka, H. (1994). An optimal-weighting criterion of case indexing for both numeric and symbolic attributes. In D.W. Aha (Ed.), *Case-based reasoning,* (Technical Report WS-94-01) (pp. 123-127). Menlo Park, CA: AIII Press.

Murphy, P.M. (1996). *UCI repositories of machine learning and domain theories.* Retrieved November 12, 2002, from http://www.isc.uci.edu/~mlearn/MLRepository.html

Nadler, M., & Eric, P. S. (1993). *Pattern recognition engineering.* (pp. 293-294). New York: Wiley.

Pfahringer, B. (1995). Compression-based discretization of continuous attributes. In *Proc. of the 12th International Conference on Machine Learning,* (pp. 456-463). Melbourne, Australia.

Quinlan, J.R. (1986). Induction of decision trees. *Machine Learning,* 1(1), 81–106.

Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning,* CA: Morgan Kaufmann Publishers, Inc.

Rachlin, J., Simon, K., Salzberg, S., & David, W.A. (1994). Towards a better understanding of memory-based and Bayesian classifiers. In *Proceedings of the Eleventh International Machine Learning Conference,* (pp. 242-250). New Brunswick, NJ: Morgan Kaufmann.

Randall, D.W. & Tony, R.M. (1997). Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research 6*, 1-3.

Salzberg, S. (1991). A nearest hyperrectangle learning method. *Machine Learning, 6*, 277-309.

Tip, F. (1995). A survey of program slicing techniques. *Journal of Programming Languages, 3*, 121-189.

Tversky, A. (1977). Features of similarity. *Psychological Review, 84*(4), 327-352.

Vasconcelos, W.W. (2000). Slicing knowledge-based systems techniques and applications. *Knowledge Based Systems Journal, 13*, 177-198.

Weiser, M. (1984). Program slicing. *IEEE Transaction Software Engineering, SE-10* (4), 352-357.

Wettschereck, D., & Aha, D.W. (1995). Weighting Features. In *Proceedings of the 1st International Conference on CBR (ICCBR-95).* (pp. 347-358). Portugal.

Wilson, D.R., & Martinez, T.R. (1996). Value difference metrics for continuously valued attributes. In *Proceedings of the International Conference on Artificial Intelligence, Expert Systems and Neural Networks*, (pp. 11-14).

Xiaoli, Q. (1999). *A case-based reasoning system for bearing design.* (Master's Thesis), Faculty of Computer Science, Drexel University. Philadelphia, PA