مجلة العلوم البحتة والتطبيقية
Journal of Pure & Applied Sciences
www.Suj.sebhau.edu.ly ISSN 2521-9200
Received 12/01/2018 Revised 09/02/2018 Published online 30/06/2018

Applying CST on Medical Datasets

Omar A. A. Shiba
Computer Science Department, Faculty Of Information Technology, Sebha University, Libya
Corresponding author: oma.shiba@sebhau.edu.ly

Abstract An important component of many data mining projects is finding a good classification algorithm; Case Slicing Technique (CST) is a classification algorithm based on program slicing techniques is examined in solving the classification problems in medical domain. The technique is experimented with three medical datasets, Hepatitis Domain (HEPA), Heart Disease (CLEV) and Breast Cancer (BCO) datasets. The experimental results are compared with other classification algorithms, K-Nearest Neighbor (K-NN) and Naïve Bayes (NB). The experimental result shows that the slicing technique is a promising classification algorithm in solving the decision making in medical classification problem.

Keywords: classification accuracy, case slicing technique, decision making, slicing.

تطبيق تقنية تشريح الحالة (CST) على البيانات الطبية

عمر عبدالغني شيبه
قسم علوم الحاسوب- كلية تقنية المعلومات- جامعة سبها، ليبيا
للمراسلة: oma.shiba@sebhau.edu.ly

الملخص أحد العناصر المهمة في العديد من مشاريع تنقيب البيانات هو إيجاد خوارزمية تصنيف جيدة. تقنية تشريح الحالة (Case Slicing Techniqe) هي خوارزمية تصنيف مبنية على تقنيات تشريح البرنامج في لغات البرمجة، هذه التقنية تم اختبارها في حل مشاكل التصنيف في المجال الطبي. حيث تم اختبار هذه التقنية مع ثلاث مجموعات من البيانات الطبيةالمتمثلة، الهيباتيتس (HEPA)، أمراض القلب (CLEV) وسرطان الثدي (BCO). Pحيث تم مقارنة النتائج التجريبية مع خوارزميات التصنيف أخرى، K-Nearest Neighbor (K-NN) , Naïve Bayes (NB) . حيث أظهرت النتائج التجريبية أن تقنية التشريح (CST) هي خوارزمية تصنيف واعدة في حل مسائل صنع القرار في المجال الطبي.

الكلمات المفتاحية: دقة التصنيف، تقنية تشريح الحالة، صنع القرار، التشريح.

Introduction

Slicing is a method used by experienced computer programmers for abstracting from programs. Starting from a subset of a program's behavior, slicing reduces that program to a minimal form which still produces that behavior. The reduced program is called a slice.

Slicing of programs is performed with respect to some criterion. Weiser [1] proposes 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 criteria [2-5].

Program slice contains all statements that could have influenced the value of a variable of interest at some program points.

The case slicing classifier is extended to program slicing technique but 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. 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 [6], [7].

The rest of this paper is organized as follows: The case slicing technique and other related terms are reviewed in section 2. A short description of selected early related classification algorithms are featured in section 3. The experimental results and the conclusion are discussed in section 4 and section 5 respectively.

Case Slicing Technique And Related Terms

Case Slicing Technique (CST) is a classification algorithm based on program slicing techniques which helps in identifying the subset of attributes used in computing the similarity measures (needed by the classification algorithm) for more details on this algorithm see[7].

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. 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.

Conceptually, case slicing algorithm is a variation of the nearest neighbor algorithms. It compares new cases to the training cases in the data file, and computes the similarity between the new cases and training cases to classify the new cases.

1. Case Representation

In a typical supervised machine-learning task, data is represented as a table of *examples* or *instances*. Each instance is described by a fixed number of measurements, or *features*, along with a label that denotes its class. Features (sometimes called *attributes*) are typically one of two types: nominal (values are members of an unordered set), or numeric (values are real numbers). The case slicing algorithm requires a set of past cases as the input, and this data set is represented as a relational data file. Each case is a record in this data file, and consists of two parts. The first part is used as the predictors for the value of the second part which is the goal variable for more details see [8].

***2.* Calculating the weights**

In case slicing algorithm the weight of each feature in classifying the new case is calculated by using simple conditional probabilities. It assigns high weight values to features that are highly correlated with the given class using

$$w(i_a) = P(C|i_a) \quad (1)$$

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 $P(C|i_a)$ is

$$P(C|i_a) = \frac{|\text{instances containing } i_a \wedge \text{class} = C|}{|\text{instances containing } i_a|} \quad (2)$$

3. Slicing Criteria

After the weight is assigned to each feature in the case at hand and to cases in the database, then the important features which has high weights are selected as a slicing criteria.

Selected Classification Algorithms

In this section only a brief description of the selected classification methods is given here, more information can be found in [9], [10] and [11] for *K-NN* and *NB* respectively.

1. K-Nearest Neighbor (K-NN)

The basic idea of the K-Nearest Neighbor 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. Then the *K-NN* algorithm selects a case, which has the highest comparison value and retrieves it [9], [10]

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* [12] Distance in *K-NN* is defined as in equation (1) below :

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

Where w_f is the parameterized weight value assigned to feature *f* as in equation (2) below:

$$w_f = P(C|i_a) \quad (2)$$

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 $P(C|i_a)$ is defined in equation (3); and the difference between *x* and *q* can be calculated as in equation (4).

$$P(C|i_a) = \frac{|instancescontainingi_a \wedge class = C|}{|instancescontainingi_a|} \quad (3)$$

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

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

2. 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 that instance is classified to a class *v* from set of possible classes *V* according to a Maximum A Priori criterion (MAP) Naive Bayes classifier can be defined as in equation (5):

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

The conditional probabilities in the above formula are obtained from the estimates of the probability mass function using training data. This Bayes classifier minimizes the probability of classification error under the assumption that the sequence of points is independent [11].

Experimental Results

In this section the results of several practical experiments are presented to examine the performance of the case slicing algorithm and other selected classification algorithms on medical datasets.

1. Selected Data Sets

We have selected three medical datasets in this study. The datasets are Hepatitis Domain (HEPA), Cleveland Heart Disease *(CLEV),* and Breast Cancer *(BCO)* datasets. These datasets were chosen to evaluate the case slicing algorithm capabilities under controlled conditions for specific data characteristics. The datasets were drawn from the UCI-Irvine repository of machine learning databases [13]. Some characteristics of these datasets are shown in Table 1.

Table 1: Characteristics of the selected datasets. B = Boolean, C = Continuous, D = Discrete

Datasets	No. Of Data	Type & No. Of Attributes	No. Of Classes
CLEV	303	9D, 6C (15)	2
BCO	699	13B, 6C (19)	2
HEPA	155	13 B, 6 C (19)	2

2. Results and Discussion

The Hepatitis Domain (HEPA), Breast Cancer *(BCO)* and heart disease (CLEV) data sets obtained from UCI machine learning repositories and domain theories [13] are experimented, Figure 1 shows the classification accuracy achieved by the different classification algorithms.

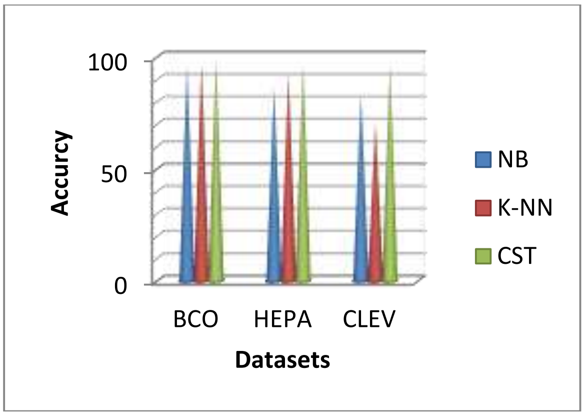

Figure 1. Comparison of the 10 folds cross-validated classification accuracies of classification techniques relative to the Case Slicing Technique (CST)

Conclusion

We have presented and discussed the Case Slicing Classification Technique (CST) in solving the classification problems in medical domain. The study tested the technique on three medical datasets. The experiment shown that using the CST indeed improves the accuracy of classification. The results are particularly good for domains which do contain many irrelevant features.

We evaluated the performance of the case slicing technique by comparing it against with K-Nearest Neighbor *(K-NN)* and Naïve Bayes (NB**)** algorithms on a three medical datasets. The datasets we have selected are very good choice to test and evaluate the slicing technique because in the selected datasets 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 remainders of the data were considered the test set. Each experiment consists of ten runs of the procedure, and the overall averages are the results reported here. The criterion of choosing the best classification approach is based on the highest percentage of classification.

References

[1]- Weiser, M. (1984), Program Slicing, IEEE Trans. Software Eng., SE-10(4), 352-357.

[2]- Agrawal, H.; and Horgan, J.R.,(1990). Dynamic Program Slicing, in Proceeding of the ACM SIGPLAN'90 Conference on Programming Language Design and implementation, New York, 246-256.

[3]- Frank Tip (1995). A Survey of Program Slicing Techniques, Journal of Programming Languages;3:. 121-189.

[4]- Wemberto W. Vasconcelos (2000), Slicing Knowledge-Based Systems Techniques and Applications, Knowledge based Systems Journal., Elsevier; 13: 177-198.

[5]- Kamkar M. (1995). An Overview and Comparative Classification of Program Slicing Techniques", J. System Software; 31:197-214.

[6]- Ming Dong, Ravi Kothari (2003). Feature Subset Selection Using a New Definition of Classifiability Computer Science Department, Wayne State University

[7]- Omar A. Shiba, Md. Nasir Sulaiman, Ali Mamat and Fatimah Ahmad, (2006). An Efficient and Effective Case Classification Method Based On Slicing, International Journal of The Computer, the Internet and Management Vol. 14.No.2, 15-23

[8]- O. Shiba (2010). Towards a Better Feature Subset Selection Approach- in proceedings of Knowledge Management 5th International Conference (KMICe 2010) 675-678

[9]- Thair N. Phyu (2009). Survey of Classification Techniques in Data Mining. In Proceeding of the International MultiConference of Engineering and Computer Scientists Vol I.

[10]- Xiaoli, Q. (1999). A Case-Based Reasoning System For Bearing Design, Master Thesis Submitted to the Faculty of Drexel University.

[11]- Tzung-I Tang, at. el.. (2005). A Comparative Study of Medical Data Classification Methods Based on Decision Tree and System Reconstruction Analysis. IEMS Vol. 4, No. 1, pp. 102-108.

[12]- Necip Fazil Ayan (1999). Using Information Gain as Feature Weight, 8th Turkish Symposium on Artificial Intelligence and Neural Networks, (TAINN'99)

[13]- Murphy, P. M.(1996). UCI Repositories of Machine Learning and Domain Theories [online]. University of California, Irvine Available: http://www.isc.uci.edu/~mlearn/MLRepository.html, [2017, Apr. 12 - Date of brows].