Assignment 2: Cluster Validation using Supervised Classifiers (VIC)

Diaz-Moraila J.E.[1], Cortes-Guzman, M.[2], and Hinojosa-Cavada, C.[3]

[1] A00828174@itesm.mx
[2] A01270966@itesm.mx
[3] A01137566@itesm.mx

Abstract. In the following work we present the construction and validation of data partitions by implementing and using Validity Index using supervised Classifiers (VIC). These partitions are created by transforming a numerical value into a categorical variable. We create 50 different binary partitions and 50 3-class partitions and report a summary of the Area Under the Curve (AUC) values found by different machine learning classifiers using the VIC framework. We present an analysis of the obtained results, reporting how the VIC evaluation changes with different partitions.

1 Introduction

Clustering algorithms have gained a lot of attention in recent years. As larger volumes of data are available, it is becoming increasingly important to make sense of this data even without human intervention. In this report we present the creation and evaluation of different partitions using Validity Index using supervised Classifiers (VIC) as presented in [1].

We create 100 different partitions in terms of a class attribute describing the score change of fingerprint minutiae classification as presented in the previous assignment. We use a categorical version of this attribute to partition the data in two or three classes which we test using VIC with 7 different machine learning algorithms: Classification and Regression Trees (CART), K-nearest Neighbors (KNN), Naive Bayes (NB), Linear Discriminant Analysis (LDA), Random Subspace Method (RSM), Logistic Regression (LR, used only in the binary problem), Adaboost (ADA, used only in the 3-class problem) and Random Forest (RF). VIC was implemented in MATLAB; the source code and documentation including example usage can be found at `https://github.com/MikeKatz45/VIC_matlab`. All the parameters of the learning algorithms were left as the MATLAB defaults except for $k = 3$ in KNN and number of iterations = 100 in RF. The MATLAB implementations of all classifiers can handle missing values except for LR (for this classifier missing values were imputed using the k-nearest neighbors method with $k = 1$).

2 Partition Algorithm

Having ordered the continuous score change values from lesser to bigger, we use an stochastic algorithm to decide the cutoff points to assign class memberships in each of the 100 partitions with a minimum cardinality of each class of 30 scores. This algorithm takes the score change array, the number of desired classes *nclass* and the minimum number of elements per class *minSize* as arguments. It identifies the minimum and maximum values in which it is possible to meet the requirements of each partitions in terms of minimum size. Once the range is identified then a random number is generated in this range to serve as a cutoff point to assign class memberships. All score change values that are between the indices of the last unassigned score and the randomly generated integer (inclusive) are assigned to a new class until all scores are assigned to one and only one class. A pseudocode of this idea would be:

```
randPart(score_change, nclass, minSize)
    n = length(score_change);

    // Initialize vector that will hold the class memberships
    part = array of length n;

    // Initialize next index to assing to a class
    next = 1;
    while nclass > 0
        nclass = nclass - 1;

        // Establish a minimum random index to be generated
        // according to the desired minimum size of each partition
        lowerlim = next + minSize - 1;

        // Establish a maximum random index to be generated
        // according to the minimum size of each partition and
        // the partitions left to assign
        upperlim = n - (minSize * nclass);

        // If this is not the last partition then generate
        // a random number in the possible range
        if nclass > 0
            i = randint(lowerlim, upperlim);
        else // assign class to all remaining indices
            i = n;

        // assign class memberships
        part[next:i] = nclass;
        next = i + 1;
    end
```

```
return part
```

We generate 50 binary partitions and 50 3-class partitions this way by varying the random number generator seed each time this algorithm is ran.

3 VIC Runs Results

This section presents the results obtained from several VIC runs with two and three classes per partition using seven different supervised machine learning classifiers.

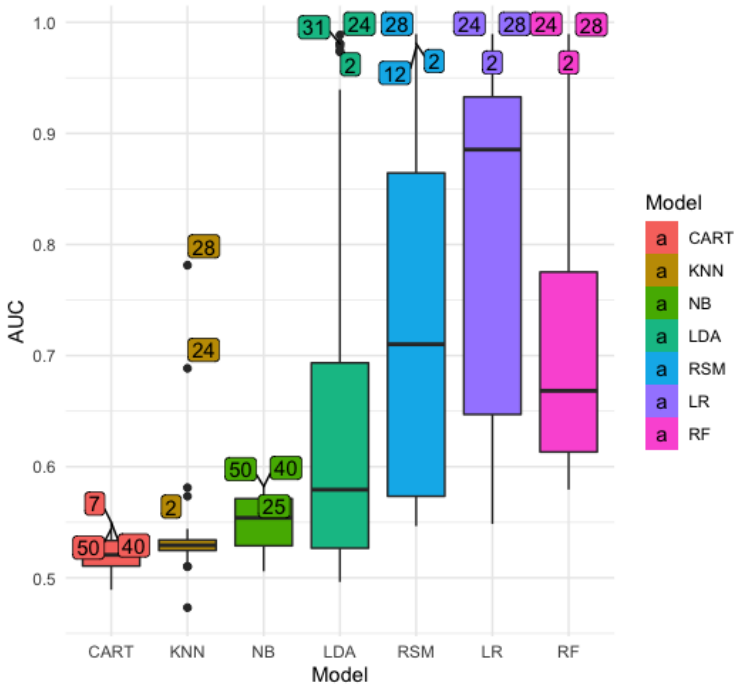

Fig. 1. AUC boxplots of 50 2-class partitions (labels depict the top 3 AUC score partitions IDs in which each classifier performed the best).

In Figure 1, it can be seen that the best classifier across all 50 partitions when examining the medians was LR. The most variable performing classifiers across the 50 partitions were LR and RSM although the more stable ones (CART and KNN) were also the worst performing in general. In the best AUC values of the best performing models, labels reveal that such performances were achieved in

fundamentally the same partitions, the same way on the worst models for the most part of the labels .

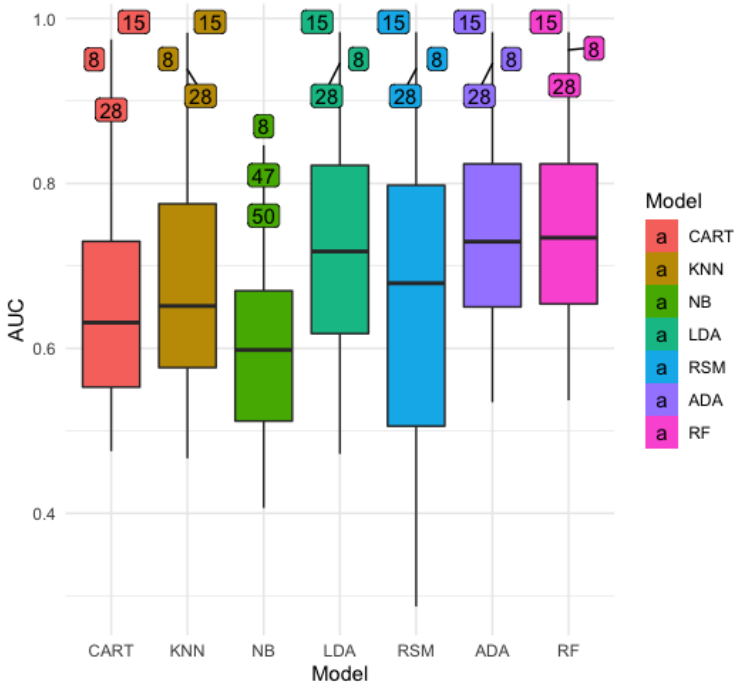

Fig. 2. AUC boxplots of 50 3-class partitions (labels depict the top 3 AUC score partition IDs in which each classifier performed the best).

For the 3-class partitions, Figure 2 reveals that in this case, median classifier performances are more even. The best median performance across the 50 partitions was achieved by RF although it was closely followed by ADA. We observe that for all classifiers except NB, the partitions evaluated that yielded the best AUC values were fundamentally the same.

Table 1. Details of the top 3 binary partitions with the highest median AUC values across classifiers

Partition ID	Median AUC	Class 1 Score Change	Class 2 Score change	Class 1 Cardinality	Class 2 Cardinality
2	0.884	[-3.2813, 0.2869]	[0.2870, 1.3339]	5041	200
28	0.8759	[-3.2813, 0.3613]	[0.3624, 1.3339]	5130	111
12	0.8527	[-3.2813, 0.2689]	[0.2691, 1.3339]	5004	238

The best global AUC achieved while running VIC in the binary partitions was 0.9894 which was observed when running RSM, LR and RF (tied for all 3) in partition 28.

Table 2. Details of the top 3 3-class partitions with the highest median AUC values across classifiers

Partition ID	Median AUC	Class 1 Score Change	Class 2 Score change	Class 3 Score Change	Class 1 Cardinality	Class 2 Cardinality	Class 3 Cardinality
15	0.9829	[-3.2813, 0.3576]	[0.3582, 0.4684]	[0.4687, 1.3339]	5125	66	50
8	0.9599	[-3.2813, 0.2555]	[0.2560, 0.2963]	[0.2977, 1.3339]	4976	75	191
28	0.9384	[-3.2813, -0.4576]	[-0.4573, 0.3677]	[0.3700, 1.3339]	267	4867	107

The best global AUC achieved while running VIC in the 3-class partitions was 0.9833 which was observed when running LDA, RSM and RF (tied for all 3) in partition 15.

References

1. Jorge Rodríguez, Miguel Angel Medina-Pérez, Andres Eduardo Gutierrez-Rodríguez, Raúl Monroy, and Hugo Terashima-Marín. Cluster validation using an ensemble of supervised classifiers. *Knowledge-Based Systems*, 145:134–144, April 2018.