On the Use of the Adjusted Rand Index as a Metric for Evaluating Supervised Classification

Jorge M. Santos[1] and Mark Embrechts[2]

[1] ISEP - Instituto Superior de Engenharia do Porto, Portugal
[2] Rensselaer Polytechnic Institute, Troy, New York, USA
emails:jms@isep.ipp.pt, embrem@rpi.edu

Abstract. The Adjusted Rand Index (ARI) is frequently used in cluster validation since it is a measure of agreement between two partitions: one given by the clustering process and the other defined by external criteria. In this paper we investigate the usability of this clustering validation measure in supervised classification problems by two different approaches: as a performance measure and in feature selection. Since ARI measures the relation between pairs of dataset elements not using information from classes (labels) it can be used to detect problems with the classification algorithm specially when combined with *conventional* performance measures. Instead, if we use the class information, we can apply ARI also to perform feature selection. We present the results of several experiments where we have applied ARI both as a performance measure and for feature selection showing the validity of this index for the given tasks.

1 Introduction

One of the main difficulties in classification problems consists on the correct evaluation of the classifier performance. This is usually done by applying a common performance measure like the Mean Squared Error (MSE) or the Classification Correct Rate. Other measures like AUC (area in percentage under the Receiver Operating Characteristic (ROC) curve), Sensitivity and Specificity, are also used specially for two class problems like those involving medical applications. All these measures *compare* the labeled outcome of the supervised classification algorithm with the known labeled targets. By doing this they evaluate how good the algorithm has labeled the input data according to the required target labels. This can lead to poor results derived only by the fact that the output labels could be switched even if the classes are well identified. In these cases we deemed useful the introduction of a measure that can evaluate how well the algorithm split the input data in different classes by looking at the relation between elements of each class and not to the given labels. This is the main reason for our proposal of using a clustering validation measure in supervised classification problems.

Usually, as we will show on the experiments, the ARI performs in a similar way as other common measures. Lower values for bad classification results and higher values for good classification results. We advise to include ARI in the set of performance measures usually used on the evaluation of supervised classification algorithms.

Since ARI is a measure of agreement between partitions and the target data is partitioned by means of the labeling we can also use ARI to perform feature selection if we split each feature in non-overlapping equal intervals and compare the partition derived from the split with the one given by the targets. By doing this we are evaluating each feature's discriminant power and we can rank the features according to the computed ARI value. We can then select the most discriminant features to apply in our classification algorithm.

This work is organized as follows: the next section introduces the Adjusted Rand Index; Section 3 explains how we intend to use ARI as a performance measure for supervised classification problems and for feature selection; Section 4 presents several experiments that show the applicability of the proposed measure with results detailed in Section 5. In the final section we draw some conclusions about the paper.

2 The Adjusted Rand Index

There are several performance indices for cluster evaluation. Indices are measures of correspondence between two partitions of the same data and are based on how pairs of objects are classified in a contingency table.

Consider a set of n objects $S = \{O_1, O_2, ..., O_n\}$ and suppose that $U = \{u_1, u_2, ..., u_R\}$ and $V = \{v_1, v_2, ..., v_C\}$ represent two different partitions of the objects in S such that $\cup_{i=1}^{R} u_i = S = \cup_{j=1}^{C} v_j$ and $u_i \cap u_{i'} = \emptyset = v_j \cap v_{j'}$ for $1 \leq i \neq i' \leq R$ and $1 \leq j \neq j' \leq C$. Given two partitions, U and V, with R and C subsets, respectively, the contingency Table 1 can be formed to indicate group overlap between U and V.

Table 1. Contingency Table for Comparing Partitions U and V.

Partition		V				
	Group	v_1	v_2	$\cdots$	v_C	Total
	u_1	t_{11}	t_{12}	$\cdots$	t_{1C}	$t_{1.}$
U	u_2	t_{21}	t_{22}	$\cdots$	t_{2C}	$t_{2.}$
	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
	u_R	t_{R1}	t_{R2}	$\cdots$	t_{RC}	$t_{R.}$
Total		$t_{.1}$	$t_{.2}$	$\cdots$	$t_{.C}$	$t_{..} = n$

In Table 1, a generic entry, t_{rc}, represents the number of objects that were classified in the rth subset of partition R and in the cth subset of partition C. From the total number of possible combinations of pairs $\binom{n}{2}$ from a given set we can represent the results in four different types of pairs:

a - objects in a pair are placed in the same group in U and in the same group in V;
b - objects in a pair are placed in the same group in U and in different groups in V;
c - objects in a pair are placed in the same group in V and in different groups in U and;
d - objects in a pair are placed in different groups in U and in different groups in V.

This leads to an alternative representation of Table 1 as a 2 × 2 contingency table (Table 2) based on a, b, c, and d.

Table 2. Simplified 2 × 2 Contingency Table for Comparing Partitions U and V.

Partition	V	
U	Pair in same group	Pair in different groups
Pair in same group	a	b
Pair in different groups	c	d

The values of the four cells in Table 2 can be calculated using the values of Table 1 by:

$$a = \sum_{r=1}^{R}\sum_{c=1}^{C}\binom{t_{rc}}{2} = \left(\sum_{r=1}^{R}\sum_{c=1}^{C} t_{rc}^2 - n\right)/2 \tag{1}$$

$$b = \sum_{r=1}^{R}\binom{t_{r.}}{2} - a = \left(\sum_{r=1}^{R} t_{r.}^2 - \sum_{r=1}^{R}\sum_{c=1}^{C} t_{rc}^2\right)/2 \tag{2}$$

$$c = \sum_{c=1}^{C}\binom{t_{.c}}{2} - a = \left(\sum_{c=1}^{C} t_{.c}^2 - \sum_{r=1}^{R}\sum_{c=1}^{C} t_{rc}^2\right)/2 \tag{3}$$

$$\begin{aligned} d = \binom{n}{2} - a - b - c &= \binom{n}{2} - \sum_{r=1}^{R}\binom{t_{r.}}{2} - \sum_{c=1}^{C}\binom{t_{.c}}{2} + a \\ &= \left(\sum_{r=1}^{R}\sum_{c=1}^{C} t_{rc}^2 + n^2 - \sum_{r=1}^{R} t_{r.}^2 - \sum_{c=1}^{C} t_{.c}^2\right)/2 \end{aligned} \tag{4}$$

where t_{rc} represents each element of the $R \times C$ matrix of Table 1.

Using these four values we are able to compute several performance indices that we will present in the following paragraphs.

Together with the well known Jaccard Index [1], the Rand Index (RI), proposed by Rand [2], was, and still is, a popular index and probably the most used for cluster validation. Rand Index can be easily computed by:

$$RI = \frac{a+d}{a+b+c+d} \tag{5}$$

and it basically weights those objects that were classified together and apart in both U and V. There are some known problems with RI such as the fact that the expected value of the RI of two random partitions does not take a constant value (say zero) or that the Rand statistic approaches its upper limit of unity as the number of clusters increases. With the intention to overcame these limitations researchers have created several different measures. Examples are the Fowlkes-Mallows [3] Index $(a/\sqrt{(a+b)(a+c)})$ or the Adjusted Rand Index (ARI) proposed by Hubert and Arabie [4] as an improvement of RI. In fact ARI became one of the most successful cluster validation indices and in [5] it is recommended as the index of choice for measuring agreement between two partitions in clustering analysis with different numbers of clusters. ARI can be computed by

$$ARI = \frac{\binom{n}{2}(a+d) - [(a+b)(a+c)+(c+d)(b+d)]}{\binom{n}{2}^2 - [(a+b)(a+c)+(c+d)(b+d)]} \tag{6}$$

or

$$ARI = \frac{\binom{n}{2}\sum_{r=1}^{R}\sum_{c=1}^{C}\binom{t_{rc}}{2} - \left[\sum_{r=1}^{R}\binom{t_{r.}}{2}\sum_{c=1}^{C}\binom{t_{.c}}{2}\right]}{\frac{1}{2}\binom{n}{2}\left[\sum_{r=1}^{R}\binom{t_{r.}}{2}+\sum_{c=1}^{C}\binom{t_{.c}}{2}\right] - \left[\sum_{r=1}^{R}\binom{t_{r.}}{2}\sum_{c=1}^{C}\binom{t_{.c}}{2}\right]} \tag{7}$$

with expected value zero and maximum value 1.

3 Using ARI as a Performance Measure and for Feature Selection

When using classification algorithms one must use performance measures to evaluate the classification results. There are some well known performance measures with their inherent advantages and drawbacks. For a detailed comparison of performance measures for classification please refer to [6].

The simple use of the classification correct rate in percentage (COR) may lead to erroneous conclusions specially if we are dealing with unbalanced data sets. Consider the case of a two-class problem with one class having 90% of the cases. If all the outputs of the classification algorithm are from the majority class we will get a COR value of 90 that can be misleading specially if one intends to detect and classify the minority class (e.g. medical applications), therefore one should be aware that special care must be taken when using COR in problems with low representative classes.

There are some performance measures specially suited for two-class problems that one must definitely use when working with these kind of datasets. Examples of these measures are:

- AUC: The area in percentage under the Receiver Operating Characteristic (ROC) curve, which measures the trade-off between sensitivity and specificity in two-class decision tables [7]. The higher the area the better is the decision rule.

- BCR: The balanced correct rate defined as $50\frac{a}{a+b} + 50\frac{d}{c+d}$ in percentage.

These two measures are based on the resulting 2×2 decision table, considering as abnormal class the one with lesser cases. They are specially suitable for unbalanced datasets where an optimistically high COR could arise from a too high sensitivity or specificity. AUC and BCR give an adequate picture in those situations.

The same way we use BCR or AUC for two-class unbalanced datasets we can also use ARI for unbalanced datasets with any number of classes. By analyzing each pair of elements ARI will measure not only the correct separation of elements belonging to different classes but also the relation between elements of the same class. In a certain way this measure pays more attention to the relation between elements than to the relation between each element and its target label. We can say that ARI evaluates the capability of the algorithm to separate the elements belonging to different classes.

Consider we have a two-class problem with half of the data belonging to each class and we apply a classification algorithm. Suppose that the result of the classification algorithm is a classification matrix (confusion matrix) with half of the elements as False Positives and the other half as False Negatives. In this case the COR is 0% meaning that the algorithm is a total disaster in terms of classification goal but, the ARI value is 1 (maximum) meaning that the algorithm is doing the correct distinction between classes but the problem is only with the data labeling. The elements are well separated but the given labels are incorrect or there is some problem in the implementation of the algorithm (we could be facing the perfect *lying machine*!). By combining ARI with other measures we can gain valuable information about the performance of our classification algorithm.

We also used ARI to perform feature selection. Since ARI gives a measure of the agreement between partitions and in classification problems the training data is partitioned by means of the given labels we can make a partition for each feature and compare it with the one given by the labels. To do this we rank the feature values by splitting them in non-overlapping equal intervals (categories) that could be as many as the number of classes. These intervals will define the partition to use, together with the class partition, in the computation of ARI index. Let us consider a simple example just to clarify this concept. Table 3 represents the values of two features from a given dataset with 12 elements with the respective class labels. By computing the ARI value for features 1 and 2 using the partition defined by the class labels $P_c = \{\{a,b,c,d\},\{e,f,g,h\},\{i,j,k,l\}\}$ and the partition defined for each feature $P_{feat1} = \{\{a,b,c,e\},\{d,f,h,l\},\{g,i,j,k\}\}$, $P_{feat2} = \{\{e,i,j,k,l\},\{f,g,h\},\{a,b,c,d\}\}$ we can rank the features according to their ARI value. In the presented case the feature with highest ARI is feat2 and therefore is the most discriminant feature.

ARI will give us the feature's discriminant power. Having ranked the existent features we select a certain number of the most discriminant ones to use in our classification algorithm. This approach is suitable for datasets with an extremely large number of features like those related with gene expression.

Table 3. A simple example to illustrate the use of ARI for feature selection

Element	a	b	c	d	e	f	g	h	i	j	k	l
Class label	1	1	1	1	2	2	2	2	3	3	3	3
feat1	0	0.3	0.1	0.5	0.2	0.4	0.7	0.5	0.9	1	0.7	0.4
feat2	1	0.8	0.9	0.7	0.2	0.4	0.4	0.5	0	0.1	0.1	0.2

4 Experiments

In the context of using ARI as a performance measure we have performed some experiments in artificial and real-world datasets. As artificial datasets we used checkerboard datasets such as the one shown in Figure 1. Checkerboard datasets are complex, controllable and unbalanced datasets. We used two different configurations: 2×2 and 4×4 checkerboards. For each one of the configurations we built three datasets with different numbers of elements (points) but with a common characteristic: a fixed number of elements belonging to the minority class (100). The percentage of elements of this minority class is 50, 25 and 10% of the total number of elements. The names of these datasets in Table 5 have the following meaning: CheckN×N(T, p) means "checkerboard N×N dataset with a total of T elements, p% of them of the minority class".

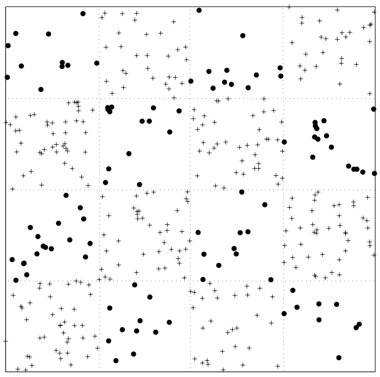

Fig. 1. An example of the 4×4 checkerboard dataset with 400 points (100 elements in the minority class: dots). Dotted lines are for visualization purpose only.

The real-world datasets are summarized in Table 4, with the top ones being the two-class datasets, the middle ones the datasets with more than two classes (multi-class problems) and the bottom ones the datasets used for feature selection. Almost every datasets can be found in the UCI repository [8] with the exception of Olive [9], Breast Tissue [10] and Leukemia [14]. The datasets differ a lot among them specially in what concerns the number of features and their topology.

Table 4. The real-world datasets.

Data set	number of elements	number of features	number of classes	number of elem. per class
Clev. Heart Disease 2	297	13	2	160-137
Diabetes	768	8	2	500-268
Ionosphere	351	34	2	225-126
Liver	345	6	2	200-145
Sonar	208	60	2	111-97
Wdbc	569	30	2	357-212
Breast Tissue	106	9	6	21-15-18-16-14-22
Clev. Heart Disease 5	297	13	5	160-54-35-35-13
Glass	214	9	6	70-76-17-13-9-29
Iris	150	4	3	50-50-50
Wine	178	13	3	59-71-48
Leukemia	72	7129	2	47-25
Arcene	100	10000	2	44-56

We used neural networks (MLP's) as classification algorithms in all problems and for the two-class problems we also used Support Vector Machines [11] that are known to be an excellent classifier for these kind of problems. In the experiments with MLP's we used the following architectures: as many inputs as the number of features, one hidden layer and one output layer for the two-class problems and as many outputs as the number of classes for multi-class problems. The number of hidden neurons, n_h, was chosen in order to assure a not too complex network with acceptable generalization. For that purpose we took into account the minimum number of lines needed to separate the checkerboard classes and the well-known rule of thumb $n_h = w/\epsilon$ (based on a formula given in [12]), where w is the number of weights and the expected error rate. Other MLP characteristics were chosen following [13]: all neurons use the hyperbolic tangent as activation function; as risk functional we used the MSE and as learning algorithm the backpropagation (BP) of the errors. The inputs were all pre-processed in order to standardize them to zero mean and unit variance. In all experiments we used the 2-fold cross validation method. In this method in each run half of the data set is randomly chosen for training and the other half for testing, then they are used with reverse roles (the original training set becomes the test set and vice-versa). Each experiment consisted of 20 runs of the algorithm. After the 20 runs the mean and standard deviation of the following performance measures were computed: AUC, COR, BCR, and ARI for the two-class problems; COR and ARI for the multi-class problems.

In the context of using ARI for feature selection we performed exploratory experiments in two data sets: a Mass-spectrometric Data for detecting cancer and; a Microarray Gene Expression Data for detecting Leukemia referred in Table 4 as Arcene and Leukemia respectively. In both experiments we used several different values for the number of intervals (categories) to split each feature and we find better results when choosing values for the number of intervals around the double of the number of classes. We selected 50 features from the 10000 of Arcene and 15 features from the 7129 of Leukemia. We have applied a Naive Bayes classifier in both cases.

5 Results

In Table 5 we show the mean and standard deviation (in brackets) of the several performance measures for the performed experiments with two-class and multi-class problems and the results for the feature selection data sets. In multi-class problems we only computed the COR, BCR and ARI performance measures since AUC is mainly for two-class problems (we also compute AUC in our daily experiments with multi-class problems since it can be obtained from the confusion matrix, however in that kind of problems it has a different meaning, reason for not showing AUC in the results because it's not appropriate for the presented comparison).

The results for the multi-class problems show a straight correlation between ARI and the *traditional* indices, specially BCR, a more reliable performance measure. The results for the Glass dataset deserve a special attention. We can see that the ARI value is more related with BCR than with COR. This is due to the characteristics of this dataset. This is a highly unbalanced dataset and by analyzing the confusion matrices (due to lack of space we do not show here the confusion matrices) we can see that the predictions are mainly restricted to 3 classes (classes 1,2 and 6) reason for the different ARI value. The results show that ARI is a good performance measure for multi-class problems.

The results of the two-class problems clearly show that ARI also gives valuable information regarding the performance of the classification algorithms. Higher values of ARI are related with higher values of the other indices. The extremely small ARI values for Liver dataset clearly points to a very complex dataset with extremely overlapping classes. When analyzing the confusion matrices we see that there are an extremely high number of misclassified elements (almost 40% of the data). These are the situations where the ARI values are smaller. Results for Diabetes also present some of this behavior.

We also can see that the ARI results for the SVM are always lower than the ones for MLP. We do not have an explanation for this, specially considering that the other performance measures do not show this same behavior.

In the feature selection problems the results for Leukemia are better than those published in [14] and for Arcene the results are not as good as those reported but we were not able to get access to all the data to perform a fair comparison. However we think that these results are very promising.

Table 5. The results with real-world and artificial datasets.

Dataset		Performance Measures			
Two-class		AUC	COR	BCR	ARI
Cleaveland HD 2	MLP	0.89 (0.01)	82.42 (1.08)	82.12 (1.03)	0.36 (0.03)
	SVM	0.90 (0.01)	83.31 (0.97)	82.90 (0.96)	0.18 (0.02)
Diabetes	MLP	0.83 (0.01)	76.58 (0.88)	72.44 (0.92)	0.20 (0.01)
	SVM	0.82 (0.01)	75.34 (2.01)	67.85 (2.91)	0.15 (0.01)
Ionosphere	MLP	0.90 (0.02)	87.81 (1.21)	84.05 (1.58)	0.56 (0.04)
	SVM	0.98 (0.01)	94.26 (0.75)	93.11 (0.85)	0.45 (0.03)
Liver	MLP	0.72 (0.02)	68.52 (1.97)	67.00 (1.98)	0.07(0.01)
	SVM	0.73 (0.02)	70.23 (2.49)	67.61 (2.76)	0.05 (0.02))
Sonar	MLP	0.89 (0.03)	78.82 (2.51)	78.59 (2.56)	0.33 (0.06)
	SVM	0.93 (0.02)	84.71 (2.01)	84.34 (2.05)	0.23 (0.03)
Wdbc	MLP	0.99 (0.001)	97.39 (0.67)	97.04 (0.71)	0.87 (0.02)
	SVM	0.99 (0.002)	96.79 (0.62)	96.43 (0.66)	0.28 (0.02)
Check2×2(1000,10)	MLP	0.63 (0.16)	95.32 (1.26)	76.97 (6.51)	0.61 (0.10)
	SVM	0.99 (0.01)	98.39 (0.49)	92.55 (2.43)	0.53 (0.03)
Check2×2(400,25)	MLP	0.96 (0.08)	95.57 (2.53)	92.47 (4.73)	0.75 (0.09)
	SVM	0.99 (0.004)	95.11 (1.13)	92.22 (1.83)	0.65 (0.03)
Check2×2(200,50)	MLP	0.98 (0.01)	92.85 (2.48)	92.87 (2.48)	0.67 (0.06)
	SVM	0.98 (0.01)	92.40 (2.09)	92.42 (2.10)	0.45 (0.05)
Check4×4(1000,10)	MLP	0.71 (0.05)	93.66 (0.76)	70.77 (3.63)	0.48 (0.06)
	SVM	0.98 (0.01)	96.04 (0.55)	82.04 (2.60)	0.27 (0.02)
Check4×4(400,25)	MLP	0.88 (0.03)	86.22 (1.61)	77.96 (3.80)	0.48 (0.06)
	SVM	0.96 (0.01)	89.70 (1.37)	84.06 (2.07)	0.35 (0.02)
Check4×4(200,50)	MLP	0.83 (0.05)	78.54 (3.80)	78.51 (3.79)	0.30 (0.07)
	SVM	0.91 (0.02)	83.18 (2.56)	83.12 (2.54)	0.24 (0.04)
Multi-class					
Breast Tissue			64.01 (3.47)	62.40 (3.47)	0.46 (0.04)
Cleaveland HD 5			58.55 (1.43)	58.55 (1.43)	0.42 (0.03)
Glass			63.13 (3.56)	53.02 (3.56)	0.29 (0.04)
Iris			96.47 (1.17)	96.17 (1.47)	0.90 (0.03)
Olive			94.19 (0.73)	94.19 (0.72)	0.90 (0.01)
Thyroid			95.19 (3.09)	92.64 (3.09)	0.84 (0.09)
Wine			97.42 (1.31)	97.42 (1.30)	0.92 (0.04)
Arcene		0.76	74.00	74.00	0.22
Leukemia		0.98	91.18	92.50	0.67

6 Conclusions

We presented and proposed in this work the use of an unsupervised classification performance measure in supervised classification problems. We have presented several experiments that show the validity of ARI index as a performance measure in classification both in two-class and multi-class datasets. We have showed that ARI is especially good for multi-class classification. By analyzing the relations between pairs of elements belonging to each predicted class and the correspondent label ARI gives valuable information about the correct separability of the classes.

We also presented two preliminary experiments that show that ARI can also be used for feature selection specially for datasets with a high number of features but we are conscious that this issue deserves a more detailed study particularly to evaluate the influence of the number of intervals (categories) in the final results.

Finally, we must say that we use this index in our daily experiments and it shows to be useful in some of them, therefore we advise all the researchers to include this index as a measure of performance of their classification algorithms.

References

1. Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. In *Bulletin del la Société Vaudoise des Sciences Naturelles*, number 37, pages 547–579. 1901.
2. W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66:846–850, 1971.
3. E. Fowlkes and C. Mallows. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, 78:553569, 1983.
4. Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.
5. Glenn Milligan and Martha Cooper. A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavioral Research*, 21:441458, 1986.
6. C. Ferri, J. Hernndez-Orallo, and R. Modroiu. An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, 30(1):27 – 38, 2009.
7. C. E. Metz. Basic principles of ROC analysis. *Seminars in Nuclear Medicine*, 8(4):283–298, 1978.
8. C. Blake, E. Keogh, and C. Merz. UCI repository of machine learning databases. http://www.ics.uci.edu/˜mlearn/MLRepository.html, 1998.
9. M. Forina and C. Armanino. Eigenvector projection and simplified non-linear mapping of fatty acid content of italian olive oils. *Ann. Chim. (Rome)*, 72:127–155, 1981.
10. Joaquim Marques de Sá. *Pattern Recognition: Concepts, Methods ans Applications.* Springer-Verlag, 2001.
11. Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
12. E. Baum and D. Haussler. What size net gives valid generalization? *Neural Computation*, 1(1):151160, 1990.
13. Christopher M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, N.Y., 1996.
14. T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.