ARL-TR-9854 ● DEC 2023

Associativity–Peakiness Metrics for Contingency Tables

by Naomi E Zirkind and William J Diehl

NOTICES

Disclaimers

ARL-TR-9854 ● DEC 2023

Associativity–Peakiness Metrics for Contingency Tables

Naomi E Zirkind and William J Diehl
DEVCOM Army Research Laboratory

REPORT DOCUMENTATION PAGE

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
December 2023	Technical Report		

4. TITLE AND SUBTITLE
Associativity–Peakiness Metrics for Contingency Tables

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**

6. AUTHOR(S)
Naomi E Zirkind and William J Diehl

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
DEVCOM Army Research Laboratory ATTN: FCDD-RLA-LD Adelphi, MD 20783-1138	ARL-TR-9854

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A. Approved for public release: distribution unlimited.

13. SUPPLEMENTARY NOTES
ORCID ID(s): Naomi Zirkind, 0009-0001-5854-8162; William Diehl, 0000-0002-6293-5018

14. ABSTRACT
For the use case of comparing the performance of clustering algorithms whose output is a contingency table, a single performance metric for contingency tables is needed. A survey of publicly available literature did not show the presence of such a metric. Metrics do exist for vector pairs of truth values and predicted values, which are an alternative form of output of clustering algorithms. These metrics could also be used to characterize the output as expressed in a contingency table, due to the interchangeability of contingency tables and the vector pairs. However, the metrics for vector pairs do not reveal the presence of detailed performance features that are apparent in contingency tables. This report presents the Associativity–Peakiness (AP) metric, which characterizes aspects of clustering algorithm performance that are critical for predicting a clustering algorithm's performance when deployed. The AP metric is analogous to measures of quality for confusion matrices that are outputs of supervised learning algorithms. This report presents results from simulations in which 500 contingency tables were generated for multiple test scenarios. The results show that for the use case of evaluating clustering algorithms, the AP metric characterizes performance of contingency tables with higher dynamic range than publicly available metrics, and that these metrics do not correlate well with the AP metric.

15. SUBJECT TERMS
Electromagnetic Spectrum Sciences, contingency tables, clustering algorithms, metrics for contingency tables, metrics for unsupervised learning, associativity of contingency tables

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UU	60

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Naomi Zirkind	(301) 394-1530

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

1. Introduction

This work addresses the need for a new metric for evaluating the performance of unsupervised clustering algorithms. Deficiencies of currently used metrics for this purpose are presented, and a novel metric is presented: associativity combined with peakiness, the AP metric.

Contingency tables and truth vector–prediction vector pairs are equivalent (assuming row ordering in the vector pairs does not matter). Demonstration of this fact is presented in the appendix. This equivalence is analogous to the duality between time domain and frequency domain representations of a signal.

However, each domain requires its own metrics. For example, bandwidth applies to a frequency-domain representation. If, for a particular use case, the time domain was most relevant, then time domain metrics, such as rise time for a pulse, must be defined. In such a case, transforming a time-domain function into the frequency domain and then using frequency-domain metrics on that function would give minimal insight into the rise-time performance of the function.

The metrics presented here are designed specifically for contingency tables. When the contingency table representation of the data is most relevant for one's use case, then metrics for the contingency table must be defined.

The AP metric is quite similar to the figures of merit for a confusion matrix, which is a primary performance metric for supervised learning algorithms. Desirable characteristics of confusion matrices are that the diagonal elements are quite large, and that all off-diagonal elements are quite small. Contingency tables will not, in general, be diagonal for two reasons: (1) they are not necessarily square matrices, and (2) their two axes are not equal as is the case with confusion matrices. Therefore, figures of merit for confusion matrices cannot be applied as-is to contingency tables. The corresponding desirable characteristics for contingency tables are that they should contain N large numbers, where N is the number of truth classes, such that each of these N large numbers corresponds to a different cluster. Also, the other matrix elements besides these N large numbers should be quite small. The first of these characteristics is measured by the associativity metric, and the second of these characteristics is measured by the peakiness metric.

The remainder of the report is organized as follows. The use case of characterizing the output of clustering algorithms is presented, with a particular contingency table presented as an example of the need for a metric that quantifies associativity and peakiness. The next section summarizes the results of a literature search for relevant metrics. It presents descriptions and equations for metrics from the Python scikit-learn library[1] and shows the scores that the scikit-learn and AP metrics give to the particular contingency table that was presented.

A detailed description is then presented of the equations for the AP metric, and of the methodology of evaluating the various metrics under test. Six test cases are described, and the performance of all the metrics is presented in each test case. The first two test cases consist of individual contingency tables that represent ideal performance and worst-case performance. In the remaining four test cases, score histograms and scatter plots are presented as results of simulations in which 500 contingency tables were generated. Also, correlation coefficients are presented that show the correlation of the existing metrics with the AP metric. The correlation coefficients show to what extent the other metrics under test characterize associativity and peakiness. The four test cases employ contingency tables of various clustering performance levels. In these four test cases, various shapes and sizes of contingency tables are used—square as well as non-square.

The appendix demonstrates the interchangeability between vector pairs of truth values and predicted values, and contingency tables.

2. Use Case

In machine learning, implementation of a supervised learning algorithm is divided into two main phases: training and inference. During the training phase, the parameters of the algorithm are optimized so that the algorithm is well fitted to the training data. Once the algorithm is optimized, the inference phase can commence, in which case a new sample is input into the algorithm, and the algorithm outputs a prediction regarding the class and/or value of the input.

Unsupervised learning has analogous phases, namely a research phase and a deployment phase. During the research phase, the researcher evaluates the performance of various algorithms, using metrics suited to the particular use case, to see which one performs best using one or more experimental datasets. Once the best-performing algorithm is identified, it is deployed to the situation in which it would be used, and it performs its unsupervised learning tasks, such as clustering, on data that is input into it.

This report addresses metrics to be used during the research phase of unsupervised learning, as described previously. Research efforts require performance metrics that would quantify desirable performance of an algorithm under test, according to the use case for which the algorithm is being developed.

The use case addressed here is the selection and/or development of an algorithm that will cluster unrecognized data input samples into clusters composed of similar data samples. During the research phase, labeled data samples are input into the candidate algorithms, though the algorithms are not informed of the labels during testing. The labels are retained only for calculating the performance metrics for each algorithm. Favorable performance of an algorithm would occur when the data clusters produced from an input dataset closely resemble the truth clusters that would be formed using the truth labels of each data sample. An ideal metric for such a research effort is one that quantifies this resemblance.

The contingency table is an informative way to summarize the output of a clustering algorithm. The axes of a contingency table from unsupervised learning are truth classes and cluster indices. The contingency table is especially suitable for visualizing and quantifying the resemblance of the formed clusters with the truth clusters. The metrics presented in this report are derived from the entries in a contingency table.

An example of a contingency table that shows the output of a particular clustering algorithm that was given a particular data set to cluster is shown in Fig. 1.

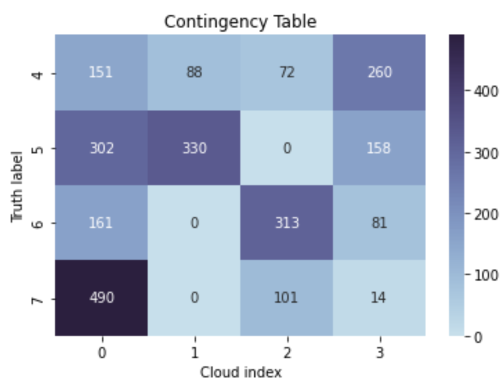

Fig. 1 Sample contingency table

The table entry squares in Fig. 1 are color coded such that the squares with higher population values are a darker color. With confusion matrices from supervised learning, one would look for a diagonal matrix. What would one look for in a contingency table from unsupervised learning? Certainly not a diagonal matrix, since the two axis labels of the table, "truth label" and "cloud index," are not equal. A close associativity between clouds and truth labels would be desirable. Looking at the darkest-colored squares in Fig. 1 shows that truth label 4 is closely associated with cloud 3, truth label 5 is closely associated with cloud 1, truth label 6 is closely associated with cloud 2, and truth label 7 is closely associated with cloud 0. Figure 2 shows the contingency table of Fig. 1 with each high-associativity square enclosed in a yellow box.

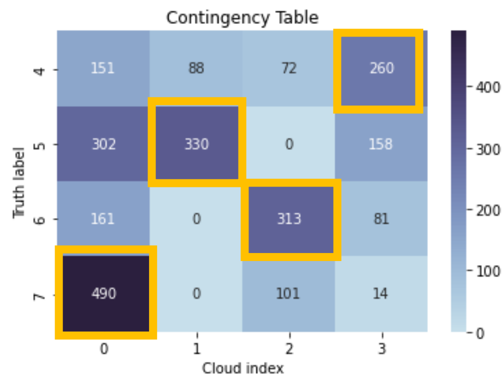

Fig. 2 Contingency table with high-associativity squares highlighted

Figure 2 shows a one-to-one associativity between clouds and truth labels. The question now becomes how to quantify the degree of associativity present in a contingency table. Furthermore, a way must be found to characterize how distinct the peak in each boxed value is in Fig. 2. The values of both the associativity and peakiness metrics must range from 0 to 1.

This report presents novel metrics for contingency tables that resemble the metrics for confusion matrices used in supervised learning. In an ideal confusion matrix, all diagonal elements are large values, and all off-diagonal elements are very small or zero. In a contingency table, the large-valued elements are not necessarily on the diagonal, but they should indicate a one-to-one matching as a confusion matrix does. The degree of one-to-one matching is quantified by the associativity metric. Furthermore, these large-valued metrics must be large compared with the other elements in the contingency table, as they are in a confusion matrix. This latter property is quantified by the peakiness metric. Thus, the AP metric combination captures the fundamental properties of a confusion matrix, which is a critical performance metric in supervised learning.

The next section surveys published literature that is relevant to the goals of finding clustering metrics in the domain of contingency tables.

3. Relevant Research Results

Much literature has been published regarding associativity for contingency tables, but these works deal with different use cases than the one considered here, which is comparatively evaluating clustering algorithms. This section summarizes some of the less similar use cases as well as a few highly similar use cases.

Two basic papers[2,3] were published by the same author in 1970. One work[2] extends previous work on association of rows and columns within a 2x2 contingency table to 2D matrices of arbitrary sizes. The other work[3] extends the existing metrics of association to multidimensional contingency tables.

A comprehensive e-book titled *Contingency Table Analysis*[4] describes a multitude of analysis techniques and metrics for 2x2 tables as well as tables with larger dimensions. These metrics address the correlation or independence between the variables whose data is tabulated in a contingency table.

Bouchet-Valat[5] evaluates various association coefficients for arbitrarily sized 2D contingency tables. The evaluation framework uses sums of rows and sums of columns, and does not consider individual indices of the table, which could be of significant interest. Silveira and Siqueira[6] give a comprehensive evaluation of many metrics for 2x2 contingency tables. Such tables measure the correlation between binary variables and do not have relevance for multi-class and multi-cluster situations.

Sayal and Kumar[7] present a metric for the similarity between attributes (rows) of a contingency table whose entries are categorical values rather than numerical values, which has been thoroughly addressed in prior works. The proposed concept for a similarity measure considers the dependency between attributes listed in the contingency table. In our use case, there is no functional dependency between truth classes or between cluster indices.

In references such as those just cited,[2–7] the contingency matrix presents results of a survey, such that the rows represent various subgroups of respondents (e.g., males and females), and the columns represent various responses to the survey. It is desired to understand the various correlations—e.g., what percent of those who responded x are males, what percent of females responded y, and so on. These approaches treat the entire contingency table as truth and try to discover correlations that emerge from this true data. In contrast, in our use case, only the class labels are truth, and we want to determine how closely the cluster distribution matches the true distribution.

A paper that is more similar to our use case is Pfitzner et al.,[8] which summarizes various pair counting, information theoretic, and entropy-based approaches to comparing two clusterings. Examples of these approaches are the Rand index and mutual information, both of which are part of the Python scikit-learn library.[1] Pfitzner et al.[8] present a set of desirable characteristics for clustering metrics and evaluate the compliance of various metrics with this set of characteristics. They propose a novel Measure of Concordance metric, which is related to the concepts of precision and recall, and show that it and a few other metrics exhibit the desired behavior in a specific set of test scenarios. However, the only connection of this work to contingency tables is that it uses a 2x2 contingency table to tabulate the results of the pair counting approaches. It does not give any performance metric for an arbitrarily sized contingency table.

Another study[9] is similar to our use case in that it compares a set of clusters produced by an algorithm with the set of "ground truth" classes. Per that study, their "evaluation scheme quantitatively measures how useful the cluster labels are as predictors of their class labels." Dom[9] presents a use case of clustering natural language documents, where the ground-truth labels were assigned by humans. Any algorithm that can produce the same clustering as one made by humans would be highly regarded. This approach uses an entropy-based metric, which measures how useful in bits the clusters are in encoding the class labels. The paper formulates a set of desired characteristics of external metrics and shows that the presented metric satisfies them all in a set of 500 test cases. The only relevance of this work to contingency tables is that, as in Pfitzner et al.,[8] a 2x2 contingency table is used to tabulate the results of a binary performance metric. Dom reports, "A value of 0 indicates pairs that were assigned to the same class, whereas a value of 1 corresponds to pairs occurring in different classes."[9(p.138)]

3.1 F1 Clustering Metric

The most germane work to our use case is in Guo et al.,[10(p.13)] which presents two metrics for the type of contingency tables that tabulate truth classes versus cluster index. Their two metrics are called truth class accuracy* and cluster purity, the latter of which is also presented in the "External Evaluation" section of Manning et al.[11] The truth class accuracy metric has a similar equation to that of the cluster purity metric, as can be seen from Eqs. 1 and 2. Guo et al.[10(p.13)] apply a formula similar to the publicly available cluster purity metric formula to the truth class dimension of contingency tables to arrive at their truth class accuracy metric.

For a contingency matrix whose rows and columns are truth class labels and cluster indices, truth class accuracy and cluster purity are defined by Eqs. 1 and 2, respectively.

$$Truth\ Class\ Accuracy = \frac{1}{N_{truth\ classes}} * \sum_{truth\ classes} \frac{\max cluster\ population}{sum\ of\ cluster\ populations} \quad (1)$$

$$Cluster\ Purity = \frac{1}{N_{clusters}} * \sum_{clusters} \frac{\max truth\ class\ population}{sum\ of\ truth\ class\ populations} \quad (2)$$

In Eq. 1, $N_{truth\ classes}$ is the total number of truth classes. In Eq. 2, $N_{clusters}$ is the total number of clusters. Guo et al.[10] mention that these metrics are useful for

* This metric had a different name in Guo et al.,[10] but a generic name (truth class accuracy) is used here to enable public distribution of this report.

comparing the performance of clustering algorithms. Guo et al.[10] combine the truth class accuracy and the cluster purity metrics into a single metric, called F1, since both metrics are critical in evaluating performance. The F1 metric is the harmonic mean of the truth class accuracy and the cluster purity.

Although Guo et al.[10] present two metrics for contingency tables, and combine them into a single metric, the associativity and peakiness metrics presented here have advantages over those metrics, as described in Section 6.7, "Comparison of AP Metric and F1 Metric."

The literature search described in this section shows that there are many studies of metrics of contingency tables in which correlations of two or more variables is quantified, but these results have limited applicability to the use case in which the contingency table represents the results of a clustering algorithm. The works cited in which a contingency table is used in connection with clustering algorithms use only 2x2 contingency tables to represent binary variables that characterize the clustering performance. Only one limited-distribution reference cited[10] presents a pair of metrics for contingency tables, but the metrics presented here are more descriptive and comprehensive than those metrics, as described in a later section.

The absence of a single, highly descriptive, publicly available metric for contingency tables is illustrated by a statement in the scikit-learn documentation. This documentation[12] lists two drawbacks of contingency tables, the second of which is, "It doesn't give a single metric to use as an objective for clustering optimization." This report presents and evaluates just such a metric.

3.2 scikit-learn Metrics

The Python scikit-learn library has a good set of metrics for unsupervised clustering in cases where truth values (also known as "ground truth") are known.[1] These metrics measure the similarity between two clusterings.

The following scikit-learn metrics are examined in this study for comparison with the novel metrics presented here: Adjusted Mutual Information (AMI), Adjusted Rand Score (ARS), Fowlkes–Mallows Score (FMS), Completeness, Homogeneity, and V-Measure. For each of these metrics, the score it produces is between 0 and 1, except that the ARS ranges from –0.5 to 1 in the scikit-learn implementation.[13] Each of these metrics is briefly described in the following paragraphs. Example calculations for all of these metrics are clearly presented in Ref. 14.

3.2.1 Adjusted Mutual Information

The Mutual Information (MI) is a measure of the similarity between two lists of labels of the same data (e.g., predicted [P] and true [T]). MI quantifies the information shared by the two clusterings and therefore can be used as a similarity measure between the two clusterings. A disadvantage of MI is that if the prediction assignments are random, a very undesirable situation, the MI score is nonzero. AMI corrects for chance; its value is 1 when the P and T are identical, and 0 when the MI equals the value due to chance alone. Equations for the metric are in the scikit-learn website[1] and in Vinh et al.[15]

3.2.2 Adjusted Rand Score

The Rand Score computes a similarity measure between two clusterings (e.g., predicted [P] and true [T]). The Rand Score is calculated using two variables, a and b, which are defined[13] as follows:

> a = the number of pairs of elements that are in the same cluster in T and in the same cluster in P
>
> b = the number of pairs of elements that are in different clusters in T and in different clusters in P

Using these definitions of a and b, the unadjusted Rand Score is given by Eq. 3:

$$Rand\ Score = \frac{a+b}{N_{pairs}} \tag{3}$$

where N_{pairs} is the total number of possible pairs in the dataset. As is the case with MI, the Rand Score gives a nonzero score to random prediction assignments. The ARS[13] remedies this deficiency and is defined by Eq. 4.

$$ARS = \frac{RS - E[RS]}{max(RS) - E[RS]} \tag{4}$$

where RS is the Rand Score, and $E[RS]$ is the expected Rand Score of random prediction assignments. As mentioned, the Adjusted Rand Index ranges from –0.5 to 1 in the scikit-learn implementation.[13]

3.2.3 Fowlkes–Mallows Score

The FMS measures the correctness of a cluster assignment using the geometric mean of the pairwise precision and recall. In particular, the score is derived from three variables that quantify the true positive, false positive, and false negative in a pairwise manner as follows[16]:

$True\ Positive\ (TP)$ = the number of pairs of points that belongs to the same clusters in both the true labels and the predicted labels

$False\ Positive\ (FP)$ = the number of pairs of points that belongs to the same clusters in the true labels but not in the predicted labels

$False\ Negative\ (FN)$ = the number of pairs of points that belongs to the same clusters in the predicted labels but not in the true labels

Using these three definitions, the FM score is defined by Eq. 5.

$$FM\ Score = \frac{TP}{\sqrt{(TP+FP)*(TP+FN)}} \tag{5}$$

3.2.4 Completeness, Homogeneity, and V-Measure Score

A clustering result satisfies Completeness if all the data points that are members of a given class are elements of the same cluster. A clustering result satisfies Homogeneity if all of its clusters contain only data points that are members of a single class. The degree of Completeness or Homogeneity for a particular clustering is based on conditional entropy considerations. The equations for the Completeness Score and Homogeneity Score are in Rosesnberg and Hirschberg.[17]

The V-measure is the weighted harmonic mean between Homogeneity and Completeness[17] as shown in Eq. 6.

$$V = \frac{(1+\beta)*homogeneity*completeness}{\beta*homogeneity+completeness} \tag{6}$$

In Eq. 6, the β parameter can be used to give greater weight to Homogeneity as compared to Completeness. In the scikit-learn implementation, the default value of β is 1. In the results shown in this report, this default value is used.

4. Application of Current Metrics to Particular Contingency Table

The six clustering metrics described previously were all applied to the contingency table of Fig. 1 in order to quantify the clustering effectiveness of the algorithm that produced the data in that contingency table. The results are shown in Table 1.

Each of the scikit-learn implementations of the metrics listed in Table 1 requires a truth value–prediction vector pair as its input. Therefore, the contingency table shown in Figs. 1 and 2 had to be transformed into a truth value–prediction vector pair before calculating the values of the scikit-learn metrics. The procedure for this transformation is shown in the appendix.

Table 1 scikit-learn scores for contingency table in Fig. 1

Metric	Score
AMI	0.246
ARS	0.162
FMS	0.370
Completeness	0.230
Homogeneity	0.267
V-Measure	0.275

The scikit-learn metrics all gave low scores to this contingency table, ranging from 0.162 to 0.370 on a scale of 0 to 1. This finding shows that the scikit-learn metrics do not characterize the associativity between the clusters formed and the truth clusters, at least for this contingency table. This deficiency in the scikit-learn metrics prompted development of new metrics that would quantify this associativity between cluster number and truth class. The next section describes the procedure for calculating the novel metrics, associativity and peakiness.

5. Associativity–Peakiness Metric

The novel metrics are called the associativity metric and the peakiness metric. The associativity metric uses data from the contingency table to quantify the degree of matching between the clusters formed by the algorithm and the truth classes. The contingency table in Fig. 1 would get an associativity score of 1. In contrast, a contingency table in which all data samples are in a single cluster would get an associativity score of 0. The associativity metric quantifies the associativity in all intermediate cases as well.

The associativity metric selects, for each truth class, the cluster with the highest population, as shown in Fig. 2, in order to calculate an associativity score. An additional metric is required to show the significance of that highest population value. If that highest value was just slightly above the populations of the other clusters for that truth class, then that highest value does not have much significance, and the associativity metric score derived from that highest value also does not have much significance. On the other hand, if the population values other than that highest value were all zero, then that highest value has very high significance. The metric used to quantify the significance of the highest cluster population for each truth class is called the peakiness metric—it shows how peaky that highest population value is.

5.1 Associativity Metric

This section shows the equations for calculating the associativity metric.

Let the contingency table be denoted at a matrix $T_{i,j}$ where the rows i are the truth classes and the columns j are the cluster indices. The first step is to find, for each truth class, the cluster number with the largest value, and to form an unordered list L of these numbers as defined in Eq. 7.

$$L = \{argmax_j(T_{i,j})\} \tag{7}$$

The list L Eq. 7 is then used to form an unordered list M of all possible pairs of distinct elements of list L , as defined in Eq. 8.

$$M = \{(L_m, L_n), m \neq n\} \tag{8}$$

The associativity metric A is derived from the list M as shown in Eq. 9.

$$A = \frac{Number\ of\ pairs\ in\ M\ whose\ members\ are\ unequal}{Total\ number\ of\ pairs\ in\ M} \tag{9}$$

An example of how the associativity metric is computed for the contingency table in Fig. 1 is shown in Fig. 3.

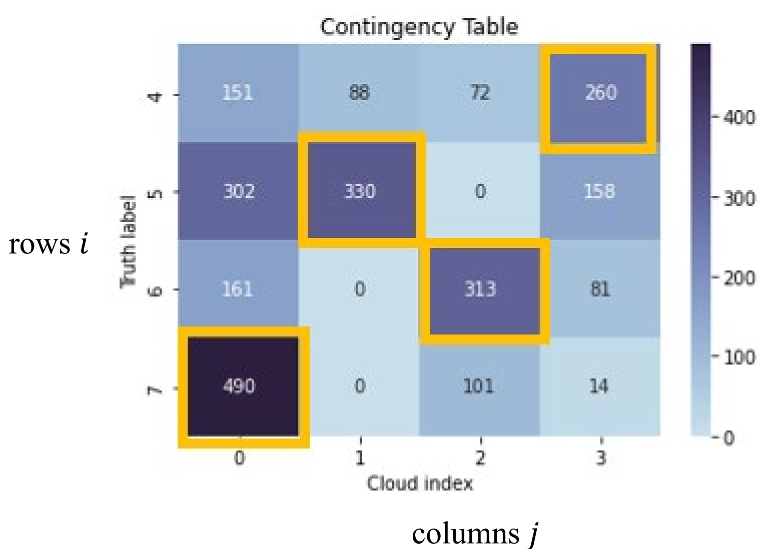

Fig. 3 Sample contingency table with rows and columns labeled

For each truth class, find the cluster number with the largest value, resulting in $L =$ [3,1,2,0].

Form the list of all possible distinct pairs of elements of L, resulting in $M =$ $[(3,1), (3,2), (3,0), (1,2), (1,0), (2,0)]$.

The number of pairs in M whose members are unequal is 6. Total number of pairs in M is 6. Therefore, the associativity metric $A = \frac{6}{6} = 1$.

5.2 Peakiness Metric

This section shows the equations for calculating the peakiness metric. This metric tells how much of an outlier are the elements of the list of L. For example, in the contingency table in Fig. 1, in the second row of this table, the maximum value of 330 is not much larger than one of the other values, 302, in the same row. Thus, the value of 330 is not much of an outlier in the elements of that row. In contrast, in the bottom row of Fig. 3, the maximum value of 490 is much larger than all the other values in that row, so it is a significant outlier. The peakiness metric tells how peaky the row maxima are, and this metric gives an indication of the confidence level of the peaks that are used to calculate the associativity metric.

When calculating the peakiness metric, the largest and second largest values will be found for each row (corresponding to a truth class) of the contingency table. The notations shown in Eqs. 10 and 11 will be used to denote the largest value and second largest value in a list N, respectively.

$$max_1(N) = \ largest\ value\ of\ list\ N \tag{10}$$

$$max_2(N) = \ second\ largest\ value\ of\ list\ N \tag{11}$$

For each truth value, the two largest values are found, and the peakiness value for the elements in row i is given by Eq. 12.

$$P_i = \frac{(\ max_1(row\ i) - \ max_2(row\ i))}{max_1(row\ i)} \tag{12}$$

Note that the expression in Eq. 12 is guaranteed to be between 0 and 1. It is 0 when the largest and second largest values in a row are equal. It is less than or equal to 1 by construction, since $max_1(L_i)$ and $max_2(L_i)$ are both nonnegative, and therefore the numerator is by definition less than or equal to the denominator.

Special care must be taken in the case when $max_1(row\ i) = 0$, which would happen only when all elements of a row are zero. In that case, the peakiness metric is undefined for that row, so that row is excluded from the calculation of the peakiness metric. A possible alternative for this case is to define the peakiness metric for that row as 0, as would be the case for any other row for which the two largest elements are equal.

The overall peakiness metric is the mean of each of the row values given by Eq. 12 and is shown in Eq. 13.

$$P = mean\ (\{P_i\}) \tag{13}$$

An example of how the peakiness metric is computed for the contingency table in Fig. 1 is shown in Table 2. For each truth class, the largest value and the second largest value are listed. The resulting calculation of the peakiness metric for each row is shown in Table 2.

Table 2 Peakiness metric calculation for sample contingency table

Truth class number	Largest value	Second largest value	P_i using Eq. 12
4	260	151	0.419
5	330	302	0.085
6	313	161	0.486
7	490	101	0.794

Note the much higher peakiness value for truth class 7 (0.794) than for truth value 5 (0.085). The overall peakiness value is the mean of all the row values $\boldsymbol{P_i}$ and is 0.446.

This section has presented the equations and sample calculations for the associativity and the peakiness metrics. The associativity metric is critical since it measures the associativity between the clusters formed by an algorithm with the truth classes. The peakiness metric is critical since it measures the confidence in the peak values used to calculate the associativity metric. Because both of these metrics are critical in quantifying the performance of an algorithm as shown in a contingency table, a composite metric is defined that combines both of these metrics into a single metric—the associativity–peakiness (AP) metric.

5.3 Associativity–Peakiness Metric

The AP metric is defined as the harmonic mean of the associativity metric (A) and the peakiness metric (P), analogous to the definition of the V-Measure metric as the harmonic mean of the Completeness and the Homogeneity metrics,[17] and the definition of the F1 metric as the harmonic mean of truth class accuracy and cluster purity as described earlier in Section 3.1, "F1 Clustering Metric." The AP metric is defined by Eq. 14.

$$AP = 2 * \frac{A * P}{A + P} \tag{14}$$

For the contingency table shown in Fig. 1, for which A = 1 and P = 0.446 as shown in the sample calculations above, Eq. 14 gives an AP value of 0.617.

Now we can compare the AP score of the contingency table of Fig. 1 with the scores of the scikit-learn metrics shown in Table 1, as well as with the F1 score, as calculated from Eqs. 1 and 2. The combined list of metrics is shown in Table 3.

Table 3 AP score and other metrics' scores for contingency table in Fig. 1

Metric	Score
AP	0.617
AMI	0.246
ARS	0.162
FMS	0.370
Completeness	0.230
Homogeneity	0.267
V-Measure	0.275
F1	0.578

Table 3 shows that for the contingency table of Fig. 1, the AP metric gives a somewhat favorable rating (0.617), whereas all the scikit-learn metrics gave very unfavorable ratings to this contingency table. The F1 metric,[10] which was designed for use with contingency tables, gave a fairly favorable rating, close to the score that the AP metric gave. Thus, in the case of the contingency table of Fig. 1, metrics designed for use with contingency tables performed better than those designed for use with truth–prediction vector pairs. These results are anecdotal; later sections compare the performance of the AP metric versus the scikit-learn metrics and the F1 metric for a much larger set of contingency tables.

6. Data Presentation

This section presents the rationale for the data presented, the methodology for specifying and generating the data, and the results that address the rationale.

When an unsupervised clustering algorithm is deployed, the only result the user obtains is the set of clusters: the number of clusters and the population of each cluster. There are some underlying truth categories to this data, but these categories are unknown to the user. The user can only assume and hope that each cluster can be associated with its unknown truth category with high confidence. The best way to ensure that this association during deployment is valid is to use a metric that measures associativity as a way to select the best-performing algorithm during the research phase. A well-designed associativity metric could support such an assurance.

The AP metric is a good candidate since it closely emulates the metrics used for confusion matrices that are used with supervised learning. The desirable features of a confusion matrix are that the diagonal elements are relatively large, and the off-diagonal elements are much smaller than the diagonal elements. In a contingency table, the most populous entries are not necessarily along the diagonal—and there might not even be a diagonal since the contingency table could be non-square. However, it is desirable that there be a one-to-one associativity between the most populous entries and the truth classes, just as there is in a confusion matrix that indicates good performance. The associativity metric quantifies this associativity. Regarding the desirable quality of the off-diagonal elements of a confusion matrix being much smaller than the diagonal elements, the peakiness metric quantifies this behavior for contingency tables. Thus, the AP metric is to contingency tables what diagonality is to confusion matrices.

The objective of presenting the data shown below in this section is to answer the following two questions:

1) What does the AP metric tell us that the scikit-learn and F1 metrics do not?
2) How well do the scikit-learn metrics and the F1 metric correlate with the AP metric (i.e., how well do they evaluate associativity)?

6.1 Evaluation Methodology

All of the performance metrics considered in this report (i.e., the AP metric, the six scikit-learn metrics, and the F1 metric) were applied to six test scenarios:

- Extreme Case – Ideal Performance
- Extreme Case – Worst-Case Performance
- Low Performance, 4x4 contingency tables
- Higher Performance, 4x4 contingency tables
- Higher Performance, 4x6 contingency tables
- Higher Performance, 4x2 contingency tables

In the first two test scenarios, a single contingency table was used. In the ideal performance case, the contingency table represents an ideal clustering result. In the worst-case performance scenario, the contingency table represents a very poor clustering result. These extreme scenarios are used to examine the basic capability of the metrics to properly characterize an extreme case. If a metric cannot correctly characterize one or both scenarios, then it cannot be effectively used for characterizing clustering performance as expressed in a contingency table.

The low performance case uses 500 randomly generated contingency tables that inherently represent low performance since the table entries were randomly generated. In this test case, a 4x4 contingency table was used.

The three higher-performance test scenarios each use 500 contingency tables that were randomly generated with a high weight placed on using zeros for the contingency table entries. The presence of these several zeros, while keeping the total of all entries fixed, forces the nonzero entries to be larger. A higher-performance contingency table has several large-valued entries, so this method of generating contingency tables more closely resembles high-performance tables as compared with tables whose entries are all randomly generated.

Three sizes of higher-performance test scenarios were used to explore the performance of the metrics with non-square contingency tables. The three table sizes are 4x4, as used in the extreme case scenarios and in the low-performance scenario, as well as 4x6 and 4x2. In each scenario, the number of truth classes is fixed at 4, and the number of clusters varies between 2 and 6.

As mentioned, for each test scenario, 500 contingency tables were randomly generated. The sum total of all elements in each contingency table is 2521, which is the sum total of all elements in the contingency table shown in Fig. 1. The number 2521 represents the total number of test samples that was used in the clustering process.

Each contingency table in test scenarios 3–6 was created by first generating a vector of random numbers with a specified number of elements and a fixed sum total (i.e., 2521). This vector was then reshaped to a 2D matrix, with dimensions as required by the test scenario.

For test scenarios 3–6, the following three data products are presented:

- Histograms of score distributions for each metric
- Bar plot of correlation coefficients between each metric and the AP metric
- Scatter plots of the scores for each metric vs. the scores for the AP metric (for scenarios 3 and 4 only)

Method for Generating the Random Vectors

Here we explain the generation of random vectors, which are reshaped to form contingency tables. The inputs to this process are the sum total of all elements in the vector (***total***) and the number of elements in each vector (***numvals***). Two Python statements,[18] Eqs. 15 and 17, were used to generate a random vector based on these two inputs.

Dividers = sorted(random.choices(range(0, total), k=numvals-1)) (15)

Equation 15 uses the Python "random" package,[19] which generates a set of pseudo-random numbers within a specified range, given a specified number of values to generate. Equation 15 creates an ordered set of $dividers$, whose form is shown in Eq. 16.

$$dividers = \{d_0, d_1, d_2, \ \dots, \ d_{numvals\ -1}\} \tag{16}$$

To generate higher-performance contingency tables, the optional argument *weights* parameter in the random.choices function was used. The *weights* parameter is a vector of relative weights for each of the elements in the $dividers$ set.[19] In this study, the value 0 was given a weight of 1000, and the value of all the rest of the elements in $dividers$ was given a weight of 1.

The second Python statement from Dickinson[18] that was used to create the random vectors is shown in Eq. 17.

Arr = [a – b for a, b in zip(dividers + [total], [0] + dividers)] (17)

The input to the **zip** function in Eq. 17 is a pair of lists that are derived from the list $dividers$ from Eq. 16. Using the notation of Eq. 16, these two lists are shown in Eqs. 18 and 19.

$$\text{List 1: } [d_0, \ d_1, \ d_2, \ \dots, \ d_{numvals-1}, total] \tag{18}$$

$$\text{List 2: } [0, \ d_0, \ d_1, \ d_2, \ \dots, \ d_{numvals-1}] \tag{19}$$

The **zip** function creates a list of corresponding pairs, one from each list, as shown in Eq. 20.

$$[(d_0, 0), (d_1, d_0), (d_2, d_1) \ \dots, (total, d_{numvals-1}) \ ] \tag{20}$$

The vector ***arr***, which is the output of Eq. 17, is a list of differences of the two elements of each list shown in Eq. 20, resulting in the expression shown in Eq. 21.

$$arr = \left[d_{0,} d_1 - \ d_0, d_2 - \ d_1, \dots, total - \ d_{numvals-1}\right] \tag{21}$$

An inspection of Eq. 21 shows that the sum of the elements in ***arr*** is indeed ***total***. The vector ***arr*** is reshaped to form a contingency table with the desired dimensions. When higher-performance contingency tables are formed, the order of the elements in ***arr*** is shuffled so that the zero values are distributed throughout the contingency table rather than all in the first one or few rows.

6.2 Results: Extreme Cases

The ideal performance for a confusion matrix from a supervised learning algorithm is a diagonal matrix. The comparable ideal performance for a 4x4 contingency table from a clustering algorithm, in which the rows are truth classes and the columns are cluster indices, is that four of the elements are nonzero, the rest of the elements are zero, and there is a one-to-one matching between the truth classes and the columns. Figure 4 displays the structure of such a contingency table that shows ideal performance.

Truth label

	x		
x			
			x
		x	

Cloud index

Fig. 4 Structure of a contingency table with ideal performance

In Fig. 4, the x's represent nonzero values, and the blank entries contain zero values. In this perfect case, the associativity metric is 1, as can be derived from Eqs. 7, 8, and 9. The associativity metric characterizes the placement of the row maxima.

In contrast to ideal performance, the worst-case performance occurs when all of the nonzero entries fall into one particular cluster, as shown in Fig. 5.

Truth label

	x		
	x		
	x		
	x		

Cloud index

Fig. 5 Structure of a contingency table with worst-case performance

In such a case, the clustering algorithm gives no information regarding the association of the data samples with the various truth classes. The associativity metric gives this situation a zero score. Since the associativity metric is zero, by construction (harmonic mean, Eq. 14), the AP metric is also zero.

Now that ideal and worst-case performance has been defined, sample contingency tables with the structures defined in Figs. 4 and 5 can be defined and evaluated by all of the metrics. Figures 6 and 7 show sample contingency tables that emulate the contingency table structures shown in Figs. 4 and 5, respectively, and that show ideal performance and worst-case performance, respectively. The sum total of all the elements in both of these metrics is 2521, which is the same sum total as the contingency table in Fig. 1.

Truth label				
	0	703	0	0
	521	0	0	0
	0	0	0	296
	0	0	1001	0
	Cloud index			

Fig. 6 Contingency matrix 1, ideal performance

Truth label				
	0	521	0	0
	0	703	0	0
	0	296	0	0
	0	1001	0	0
	Cloud index			

Fig. 7 Contingency matrix 2, worst-case performance

The scores that each of the metrics gave to the two contingency tables of Figs. 6 and 7 are shown in Table 4.

Table 4 shows that all of the metrics correctly scored the ideal case. Almost all of the scikit-learn metrics correctly scored the worst case. The AMI score was not exactly zero, at 7.0 E-16, but was very close to it. The Completeness metric gave a 1.0 score to the worst-case metric, but the V-Measure metric, which combines Completeness and Homogeneity, does give a zero score. The FMS metric gave a very poor score, 0.540. The F1 metric, which is not a scikit-learn metric, and which was designed for use with contingency tables, gave an even poorer score at 0.568. The nonzero scores for the worst case are shaded in yellow in Table 4 to highlight their presence. Inspecting the equations for the FMS (Eq. 5) and the F1 (Eqs. 1 and 2) metrics shows that their definitions would not give a zero score to the worst-case contingency table of Fig. 7.

Table 4 Scores of metrics for ideal and worst-case contingency tables

Metric	Matrix 1 score (ideal)	Matrix 2 score (worst case)
AP	1.0	0.0
AMI	1.0	0.0
ARS	1.0	0.0
Completeness	1.0	1.0
Homogeneity	1.0	0.0
V-Measure	1.0	0.0
FMS	1.0	0.540
F1	1.0	0.568

6.3 Results: Low-Performance Case

Figure 8 shows, for each of the metrics under test, the histogram of scores for the 500 randomly generated contingency tables. The metrics whose plots are shown in Fig. 8 are AMI, ARS, FMS, Completeness, Homogeneity, V-Measure, F1 Score, and the AP Score. The scores that were used to construct the histograms were derived from 4x4 contingency tables with low performance, as described in Section 6.1.

The histograms In Fig. 8 show that ARS and the entropy-based metrics, AMI, Completeness, Homogeneity, and V-Measure, gave low scores to the set of contingency tables—the most common value was about 0.15 for all of them. However, none of these metrics gave a zero score to any of the contingency tables.

On the other hand, the metrics not based on entropy, FMS and F1, gave much higher scores—their peak values were 0.4 and 0.5, respectively. Furthermore, the range of values of FMS and F1 was only about 0.3 wide, slightly narrower than the distributions from ARS and the entropy-based metrics. FMS values ranged from 0.3 to 0.6, and F1 values ranged from 0.4 to 0.7.

Only the AP metric gave some zero scores, but it also gave many higher scores such that its peak value was about 0.55 and its maximum value was greater than 0.8. None of the metrics besides AP gave such a wide range of scores. The AP metric shows a higher dynamic range, and thus a higher sensitivity, than the other metrics.

To better understand the behavior of the various metrics, especially at the extremes of high scores and zero scores, four specific contingency tables of the 500 generated samples from this test case are examined in more detail in the next section.

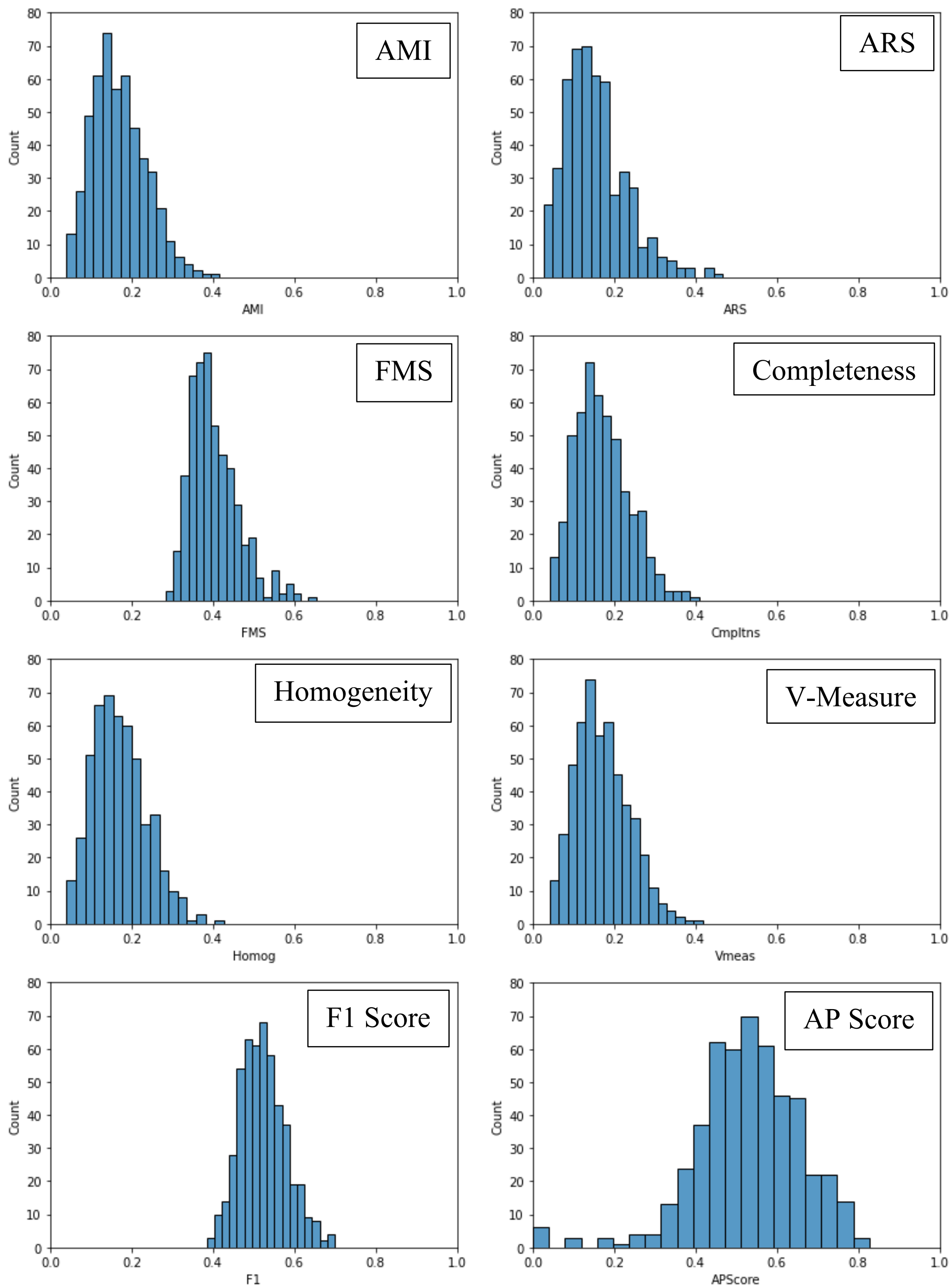

Fig. 8 Histogram distributions for metrics: low performance, 4x4

6.3.1 Analysis of Selected Contingency Tables

In this study, 500 contingency tables were generated for each test scenario. For the low-performance scenario, in which the contingency tables contained few zeros, four contingency tables were selected for inspection of their values, and for examination of scores assigned to them by the various metrics. Two of these selected contingency tables had the two highest AP scores, and two of them had AP scores of zero.

For each of the contingency tables presented, the rows are truth values, and the columns are cluster indices. For each row (i.e., truth value) in these contingency tables, the maximum value is highlighted to facilitate understanding of the scores of the metrics.

Highest AP score. Figure 9 shows the contingency table that generated the highest AP score of the entire set of contingency tables that were generated. Table 5 shows the scores that each of the metrics gave to the contingency table shown in Fig. 9.

Truth label				
	122	71	158	407
	17	66	316	7
	86	453	41	75
	351	138	117	96
	Cloud index			

Fig. 9 Sample contingency table 1, high AP score

Table 5 Metrics scores for sample contingency Table 1

Metric	**Score**
AP	0.827
AMI	0.237
ARS	0.231
FMS	0.428
Completeness	0.237
Homogeneity	0.240
V-Measure	0.238
F1	0.617

In Fig. 9, there is a one-to-one matching between clusters and truth classes, resulting in an associativity score of 1. For each row of the contingency table in Fig. 9, the peak is much larger than all the other values in that row, resulting in a high peakiness score, and thus a high AP score of 0.827.

The entropy-based metrics and ARS all gave scores of between 0.2 and 0.3 to this contingency table, except for FMS and F1, which tend to give higher scores for this dataset. However, their scores, 0.428 and 0.617, do not come near the score that the AP metric gave to this very high-performing contingency table.

Second-highest AP score. Figure 10 shows the contingency table that generated the second-highest AP score of the entire set of contingency tables that were generated. Table 6 shows the scores that each of the metrics gave to the contingency table shown in Fig. 10.

Truth label				
	106	547	18	308
	223	108	45	65
	43	14	700	27
	12	32	1	272

Cloud index

Fig. 10 Sample contingency table 2, high AP score

Table 6 Metrics scores for sample contingency Table 2

Metric	Score
AP	0.819
AMI	0.417
ARS	0.429
FMS	0.588
Completeness	0.408
Homogeneity	0.427
V-Measure	0.417
F1	0.694

The contingency table in Fig. 10 shares the good qualities of the contingency table in Fig. 9, though the peakiness scores of the first and second rows in Fig. 10 are not so high, resulting in a slightly lower AP score of 0.819 as compared to the contingency table in Fig. 9. In contrast to Table 5 for the contingency table of Fig. 9, Table 6 shows that the entropy-based metrics and ARS gave higher scores (between 0.4 and 0.5) to the Fig. 10 contingency table than they did to the Fig. 9 contingency table. The F1 metric also gave a slightly higher score (0.694 vs. 0.617) to the Fig. 10 contingency table than it did to the contingency table in Fig. 9.

AP score of zero. Figure 11 shows a contingency table that gave an AP score of zero. Table 7 shows the scores that each of the metrics gave to the contingency table shown in Fig. 11.

Truth label				
	34	100	20	15
	165	917	110	145
	123	306	27	65
	104	163	94	133

Cloud index

Fig. 11 Sample contingency table 3, zero AP score

Table 7 **Metrics scores for sample contingency table 3**

Metric	**Score**
AP	0.0
AMI	0.040
ARS	0.095
FMS	0.444
Completeness	0.042
Homogeneity	0.040
V-Measure	0.041
F1	0.502

Almost all of the metrics gave very low scores—less that 0.1—to the contingency table of Fig. 11, though only the AP metrics gave it a score of zero. The only two metrics that gave higher scores to this contingency table are FMS at 0.444, and F1 at 0.502. These two metrics do not regard the occurrence of all truth-class peaks in the same cluster, as show in Fig. 11, as an undesirable phenomenon.

Another case with AP score of zero. Figure 12 shows a second contingency table that gave an AP score of zero. Table 8 shows the scores that each of the metrics gave to the contingency table shown in Fig. 12.

Truth label / Cloud index

266	47	91	172
426	17	7	66
654	93	90	25
371	63	14	119

Fig. 12 Sample contingency table 4, zero AP score

Table 8 Metrics scores for sample contingency table 4

Metric	**Score**
AP	0.0
AMI	0.070
ARS	0.041
FMS	0.390
Completeness	0.086
Homogeneity	0.061
V-Measure	0.071
F1	0.522

As with the contingency table of Fig. 11, almost all of the metrics gave scores less than 0.1 to the contingency table of Fig. 12. The FMS metric gave a slightly lower score (0.390) to the contingency table of Fig. 12 than it did to the Fig. 11 contingency table (0.444). In contrast, the F1 metric gave a slightly higher score (0.522) to the Fig. 12 contingency table than it did to the Fig. 11 contingency table (0.502).

All of the metrics tested gave similar scores to both contingency tables that had an AP score of zero that were presented previously. It is instructive to present two different cases so that the variability in the scores from one contingency table to another can be viewed.

6.3.2 Scatter Plots and Correlation Coefficients

Figure 13 shows the scatter plots of each metric under test vs. the AP metric for 4x4 contingency table with low-performance data. On the plot for each metric, the correlation coefficient between that metric and the AP metric is shown. In each plot, the AP metric score is on the x-axis, and the score for that metric is on the y-axis.

The scatter plots in Fig. 13 show that for the contingency tables for which the AP metric gave a zero score, all but the FMS and F1 score gave these points scores less than 0.2. ARS gave especially low scores, even less than 0.1, to these points. This corresponds to the performance of these metrics in the extreme worst case described earlier in Section 6.2, "Results: Extreme Cases," where most metrics gave a zero score.

None of the metrics showed a very high correlation with the AP metric, though F1 had the highest correlation at 0.602. For the F1 score, excluding the points for which AP gave a zero score, which are enclosed in an oval in Fig. 13, the rest of the data points show quite good correlation with the AP score. An illustration of the low correlation between the metrics tested and the AP metric is that all the metrics except F1, and to some extent FMS, consistently gave lower scores than AP.

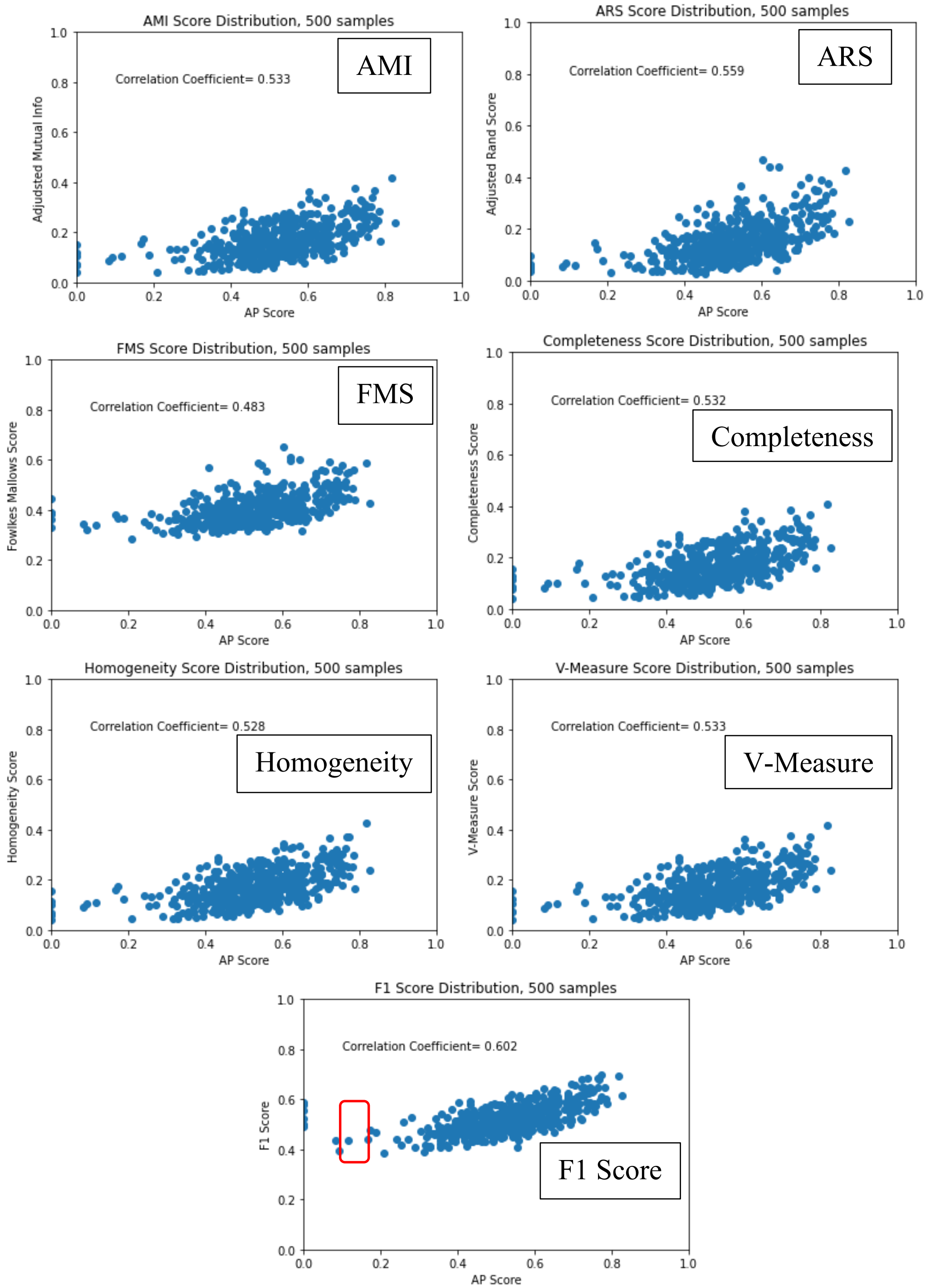

Fig. 13 Scatter plots for metrics: low performance, 4x4

Figure 14 shows a bar plot of the correlation coefficients between each metric under test and the AP metric. These correlation coefficients are displayed on each individual plot in Fig. 13 and are summarized in the plot of Fig. 14.

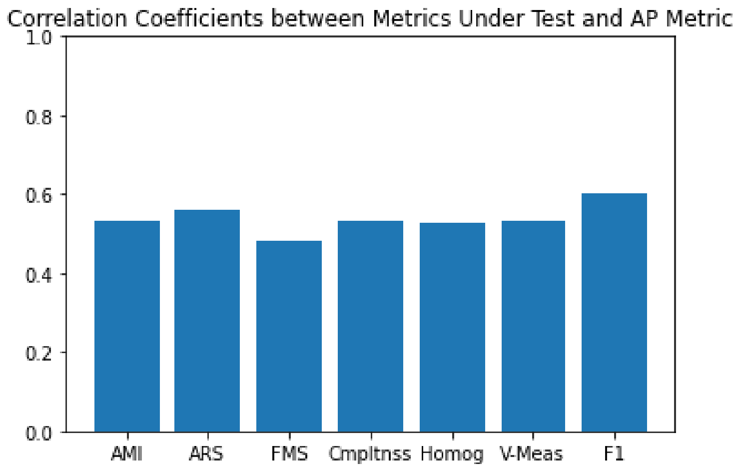

Fig. 14 Correlation coefficients: low performance, 4x4

The correlation coefficients shown in Fig. 14 range from 0.483 for FMS to 0.602 for F1. Bobbitt[20] presents a chart, shown in Table 9, that gives verbal descriptions of the strength of relationship for various values of the correlation coefficient.

Table 1 Strength of relationship for various values of correlation coefficient[20]

Absolute value of correlation coefficient r	Strength of relationship
r < 0.25	No relationship
0.25 < r < 0.5	Weak relationship
0.5 < r < 0.75	Moderate relationship
r > 0.75	Strong relationship

Based on the values in Table 9, FMS has a weak correlation with the AP metric, and the rest of the metrics have a moderate correlation with the AP metric. The scatter plot for the F1 metric in Fig. 13 shows that the F1 metric has the strongest correlation with the AP metric.

This section has presented results for contingency tables that represent poor clustering performance. The next three sections present results for contingency tables that represent better clustering performance. The method for generating the higher-performance contingency tables is described earlier in Section 6.1, “Evaluation Methodology.” The next three sections present data for different-sized contingency tables—4x4 as was considered in the present section for low-performance contingency tables, as well as 4x6 and 4x2. Each case represents situations in which the clustering algorithm produced different numbers of clusters.

6.4 Results: Higher-Performance Case, 4x4 Contingency Tables

Figure 15 shows, for each of the metrics, the histogram of scores for the 500 randomly generated 4x4 contingency tables, which represents higher clustering performance. These contingency tables would be produced by clustering algorithms that produce four clusters from data that has four truth classes.

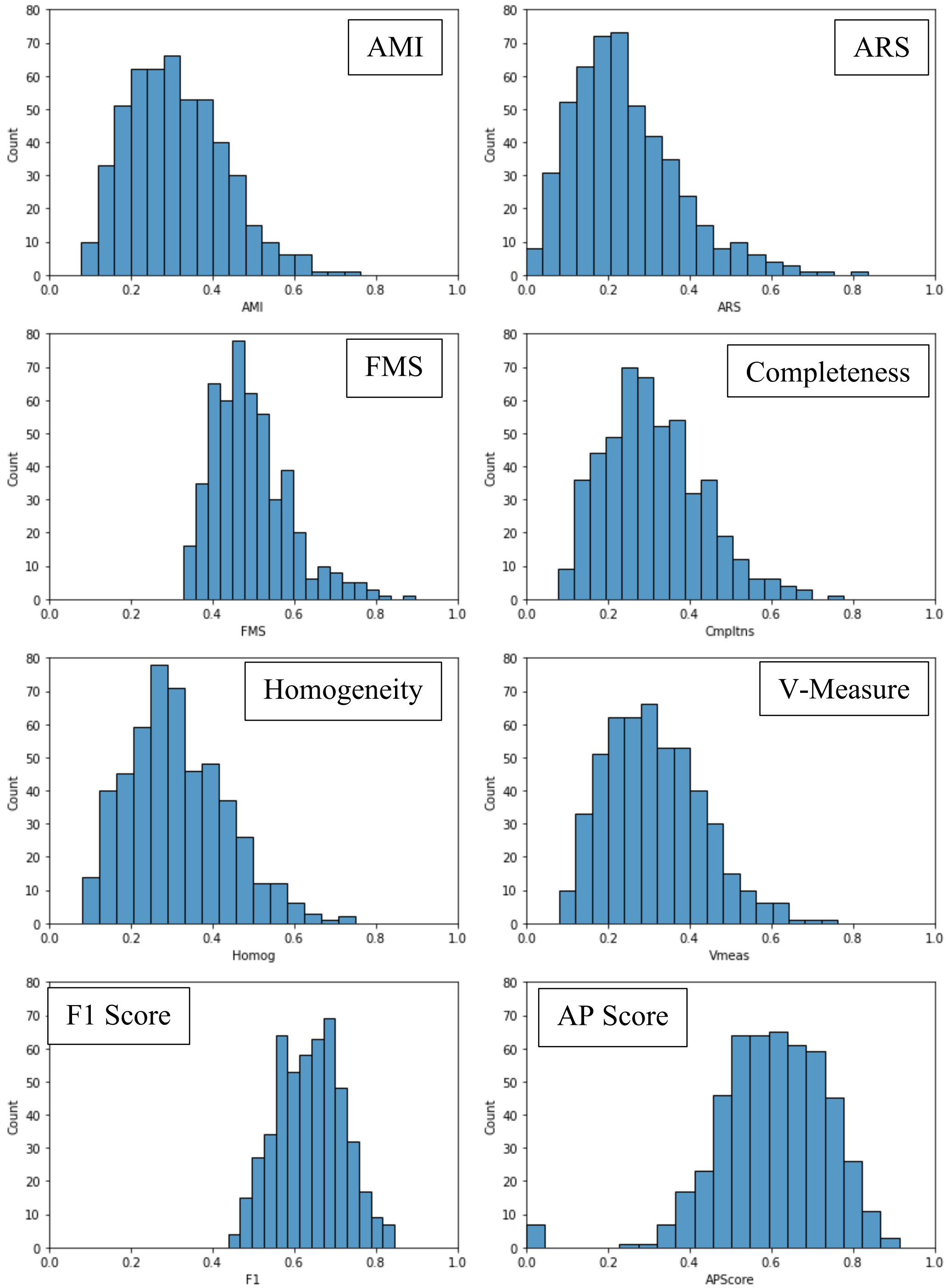

Fig. 15 Histogram distributions for metrics: higher performance, 4x4

The score distribution plots in Fig. 15 are generally shifted toward higher scores as compared with the comparable plots in Fig. 8 for the low-performance 4x4 contingency tables, confirming the designation of higher performance. The peaks' values in the scores distributions are larger, and the right-hand tails of the distributions, which represent higher scores, are shifted toward higher values. The AP score distribution behaves similarly—there are fewer very low scores, and the overall distribution is shifted toward higher scores.

Figure 16 shows the scatter plots of each metric under test vs. the AP metric. The distributions of the plots in Fig. 16 are significantly wider than the scatter plots in Fig. 13 for the lower-performing contingency tables. This effect is also visible in the histograms of the scores distributions in Fig. 15.

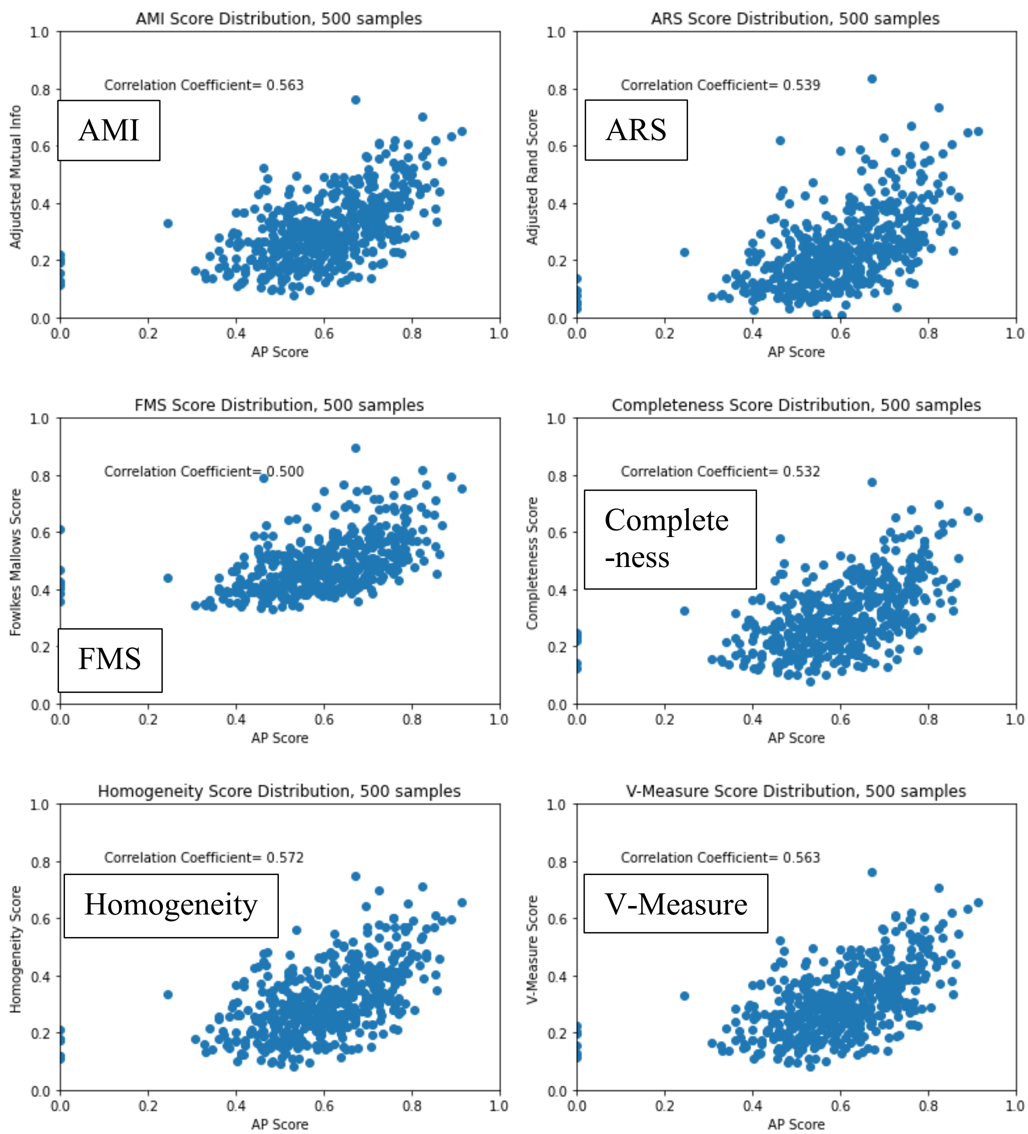

Fig. 16 Scatter plots for metrics: higher performance, 4x4 (part 1 of 2)

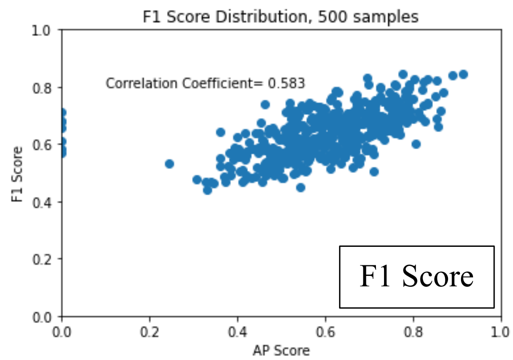

Fig. 16 Scatter plots for metrics: higher performance, 4x4 (part 2 of 2)

Figure 17 shows a bar plot of the correlation coefficients between each metric under test and the AP metric for the 4x4 contingency tables with higher clustering performance.

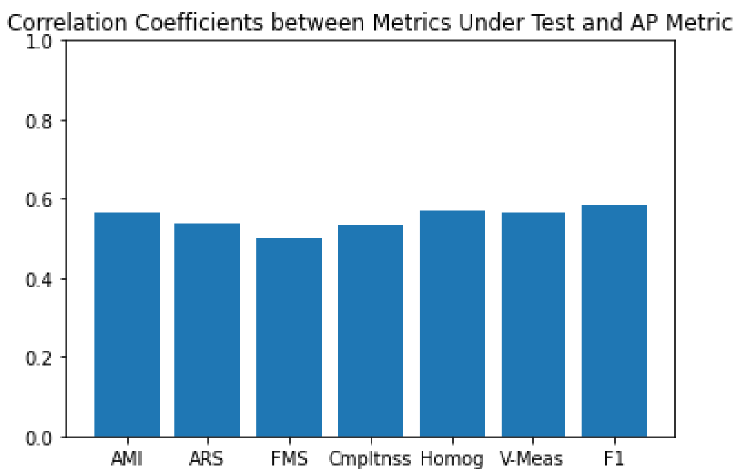

Fig. 17 Correlation coefficients: higher performance, 4x4

The correlation coefficients in Fig. 17 are similar to those in Fig. 14 for the lower-performance contingency tables. For some metrics, the score increased slightly, and for some it decreased slightly. For all metrics, the correlation coefficient reflects moderate correlation according to the designations in Table 9.

6.5 Results: Higher-Performance Case, 4x6 Contingency Tables

Figure 18 shows, for each of the metrics, the histogram of scores for the 500 randomly generated 4x6 contingency tables, which represents higher clustering performance. These contingency tables would be produced by clustering algorithms that produce six clusters from data that has four truth classes. Comparing the histogram distributions in Fig. 18 with those in Fig. 15 for the 4x4 contingency tables shows the effect of having contingency tables with a larger number of clusters.

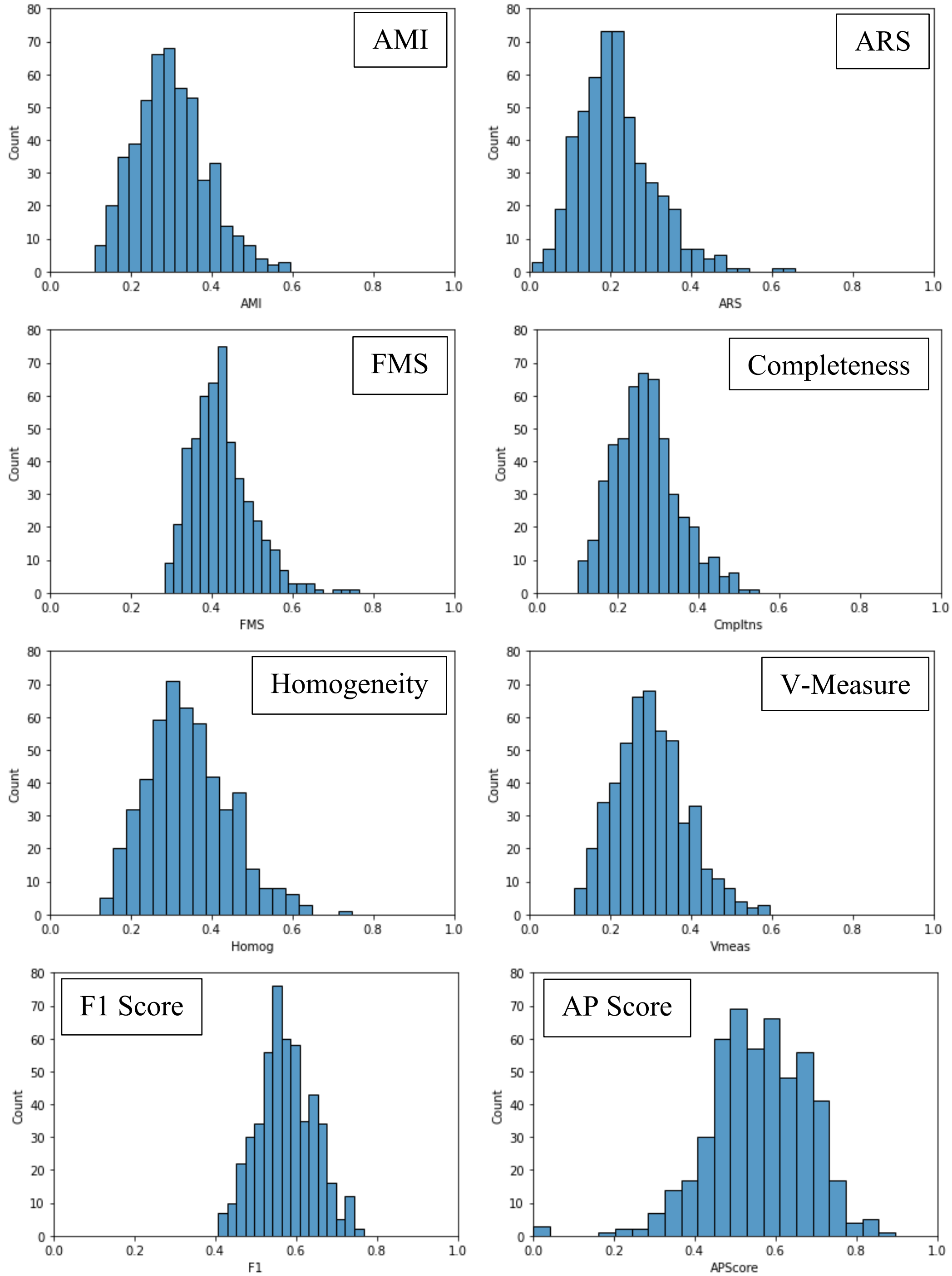

Fig. 18 Histogram distributions for metrics: higher performance, 4x6

For all of the metrics, the histogram distributions in Fig. 18 are very similar to those in Fig. 15. Thus, increasing the number of clusters produced by clustering algorithms from four to six with four truth classes does not significantly affect the distribution of the scores of any of the metrics tested here.

Figure 19 shows a bar plot of the correlation coefficients between each metric under test and the AP metric for the 4x6 contingency tables with higher clustering performance.

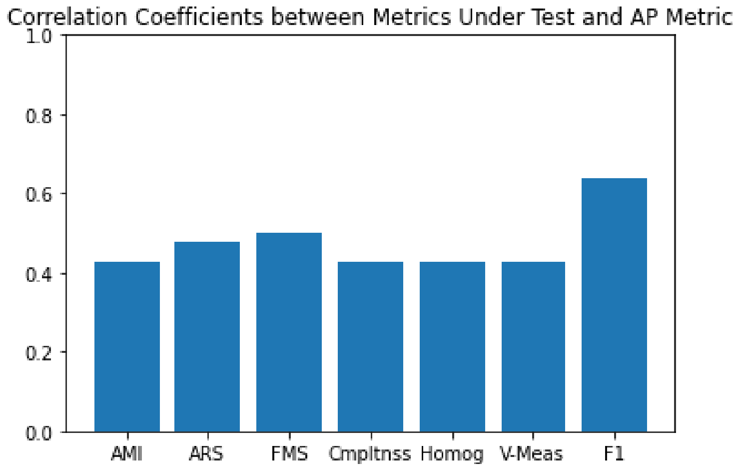

Fig. 19 Correlation coefficients: higher performance, 4x6

Comparing Fig. 19 to the comparable plot in Fig. 17 for 4x4 contingency tables shows that for all the metrics except the F1 metric, the correlation with the AP metric decreased when the contingency tables had six clusters instead of four. In contrast, the F1 metric's correlation increased from 0.583 in the 4x4 case to 0.638 in the 4x6 case. Figure 18 shows that the score distributions of the F1 metric and the AP metric are quite similar in that both have a peak of slightly less than 0.6, and they have a similar distribution of scores larger than 0.6.

6.6 Results: Higher-Performance Case, 4x2 Contingency Tables

Figure 20 shows, for each of the metrics, the histogram of scores for the 500 randomly generated 4x2 contingency tables, which represents higher clustering performance. These contingency tables would be produced by clustering algorithms that produce two clusters from data that has four truth classes.

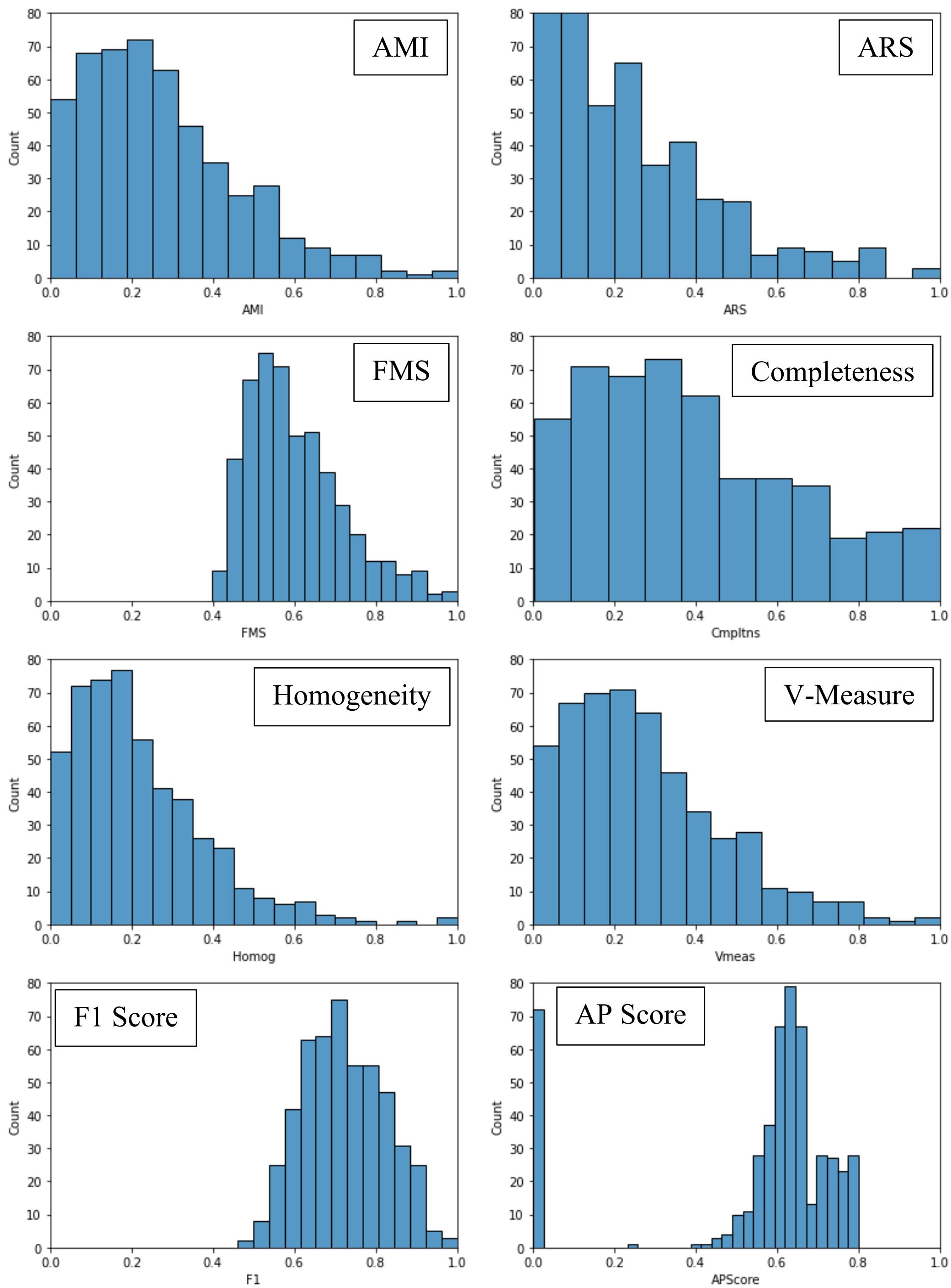

Fig. 20 Histogram distributions for metrics: higher performance, 4x2

The histograms in Fig. 20 for the 4x2 contingency tables differ significantly from the histograms in Fig. 15 for the higher-performance 4x4 contingency tables. ARS and the entropy-based metrics, AMI, Completeness, Homogeneity, and V-Measure, had many more scores at the low end of the scale, including many zero scores. That is not surprising, since with 4x2 contingency tables, many of the contingency tables would have the worst-case configuration in which all data samples are in the same cluster, as shown in Fig. 7, for example. ARS and all of the entropy-based metrics except Completeness gave a zero score to such a contingency table. Even so, Completeness did give many contingency tables a zero or near-zero score in the 4x2 case.

Another difference in the behavior of the entropy-based metrics and ARS in the 4x2 case as compared with the 4x4 case is that these metrics had a few more very high scores in the 4x2 case than in the 4x4 case.

The FMS and F1 metrics behaved quite differently than the entropy-based metrics and ARS in the comparison between the 4x4 and 4x2 contingency tables. For these two metrics, the distribution of scores changed little, except that the entire distributions shifted toward higher values.

The AP score distribution changed in yet a different way in the 4x2 case as compared with the 4x4 case. In the 4x2 case, it gave very many contingency tables a zero score, due to the prevalence of worst-case contingency matrices, as described earlier.

Notably, in the 4x2 case, only the AP score gave no scores of 1, whereas all the other metrics did give some scores of 1 or close to 1. This could be explained by the impossibility of getting a high associativity score in contingency matrices with fewer clusters than truth classes. In such cases, it is impossible for there to be a one-to-one matching of clusters with truth classes. This is an important finding, since the presence of fewer clusters than truth classes shows that class differences in the data are being obscured by the clustering algorithm, and the AP scores reflect this fact.

Figure 21 shows a bar plot of the correlation coefficients between each metric under test and the AP metric for the 4x2 contingency tables with higher clustering performance.

Comparing Fig. 21 for the 4x2 case with Fig. 17 for the 4x4 case shows that in the 4x2 case, the correlation coefficient of all of the metrics with the AP score is smaller in the 4x2 case.

For the F1 metric, the decline was especially large—from 0.583 in the 4x4 case to 0.185 in the 4x2 case. In the 4x2 case, the F1 metric gave higher scores than the AP score for the large majority of the contingency tables, as shown in Fig. 22.

This decreased correlation between the F1 and AP metrics in the 4x2 case is partly due to the many zero scores that the AP metric gave, whereas the F1 metric gave no scores even close to zero in the 4x2 case.

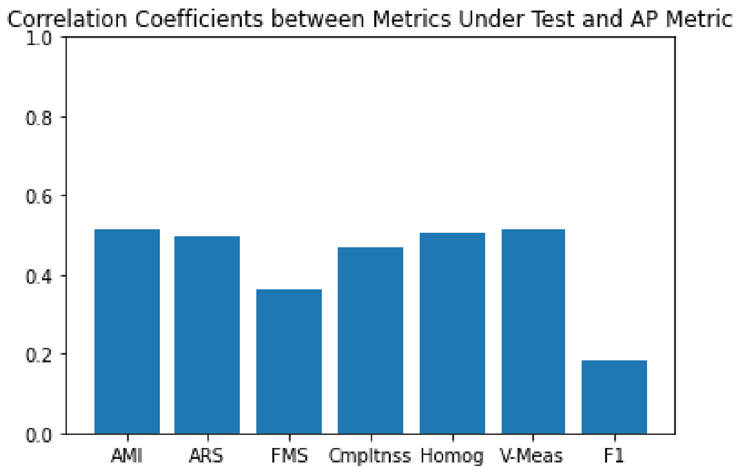

Fig. 21 Correlation coefficients: higher performance, 4x2

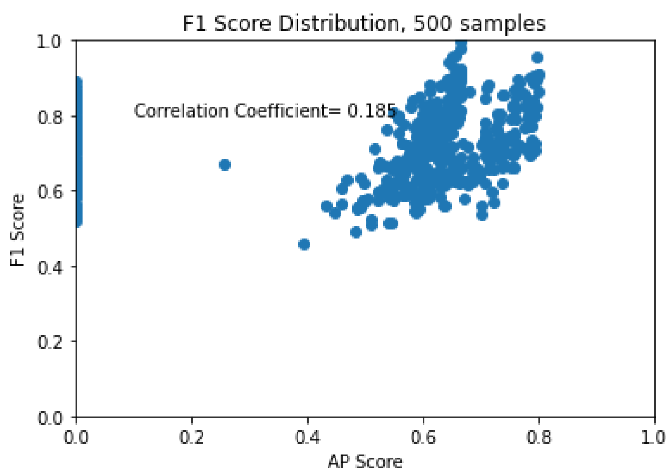

Fig. 22 Scatter plot for F1 and AP metrics: higher performance, 4x2

6.7 Comparison of AP Metric and F1 Metric

Of all the metrics under test in this study, only the F1 metric was designed for use with contingency tables in order to comparatively evaluate clustering algorithms. Furthermore, in most test scenarios, it was the most highly correlated with the AP metric. Thus, it is worthwhile to compare these two metrics in detail.

As mentioned earlier, the metrics in Guo et al.[10(p.13)] are truth class accuracy (Eq. 1) and cluster purity (Eq. 2). A disadvantage in the truth class accuracy metric becomes apparent when a clustering algorithm produces, for one or more truth values, a few clusters with large populations and several clusters with relatively small populations. In that case, even if the maximum cluster population is large for a particular truth class, the multitude of small cluster populations for that truth class can make the denominator (sum of cluster populations) quite large and thereby reduce the value of truth class accuracy. However, the presence of these multiple low-population clusters would not diminish confidence in the validity of the maximum cluster population value.

As an example, consider the contingency table of Fig. 1, which contains only the first four columns of the full contingency table, shown in Table 10.

Table 10 Contingency table of Fig. 1 with additional columns

	Cloud 1	Cloud 2	Cloud 3	Cloud 4	Cloud 5	Cloud 6	Cloud 7	Cloud 8	Cloud 9
Row 1	151	88	72	260	21	0	0	55	13
Row 2	302	330	0	158	24	0	0	9	0
Row 3	161	0	313	81	7	0	0	1	65
Row 4	490	0	101	14	82	0	0	0	4

Table 10 shows data for nine clouds; clouds 5–9 have much lower populations than clouds 1–4. According to Eq. 12 for the peakiness metric, the originally calculated values in Table 2 would be the same for the contingency table shown in Table 10, since the peakiness metric depends only on the two largest values in a row, which in this case are located in the left-most four columns (i.e., clouds 1–4) for each row.

However, according to Eq. 1 for the truth class accuracy metric, this metric is defined as the peak value in a row divided by the sum total of all values in that row, and therefore the value of the truth class accuracy metric would be affected by the presence of the data entries in clouds 5–9. Table 11 shows the scores of the peakiness metric and the truth class accuracy for each of the four rows of Table 10.

Table 11 Peakiness and truth class accuracy scores

Row	Peakiness	Truth class accuracy
1	0.419	0.394
2	0.085	0.401
3	0.486	0.498
4	0.794	0.709

Table 11 shows that the largest difference between the scores of peakiness and truth class accuracy is in row 2 of Table 10. For the truth class accuracy metric, the score of 0.401 is similar to the scores of the other three rows for that metric. However, the peakiness metric severely penalizes this row, since the second-largest value, 302, is very close to the second-largest value, 330, and therefore the confidence in

330 as the peak value for that row is low. The truth class accuracy metric does not detect the low confidence in the peak value in row 2.

The second-largest difference between the scores of peakiness and truth class accuracy is shown in row 4 of Table 11. In this row, the peakiness is quite high due to the presence of one value, 490, which is much larger than all other values in that row. In contrast, the truth class accuracy is lower than the peakiness in this row, due to the presence of several values high population values that increase total population in that row, which increases the denominator of Eq. 1.

In summary, the peakiness metric explicitly measures the extent to which the peak value in a row is an outlier compared to the rest of the values in that row. The truth class accuracy metric measures the ratio of the peak value in a row to the total population in that row. As a result, the truth class accuracy metric does not detect cases in which the peak value is just slightly larger than other value(s) in that row. Also, truth class accuracy is affected by the presence of several low-population clusters, which are not relevant to the extent to which the peak value is an outlier.

The second metric used in Guo et al.[10] is cluster purity. This metric is defined very similarly to truth class accuracy in that it seeks to reward high peaks in the truth class populations for each cluster. The combination of the two metrics, truth class accuracy and cluster purity, does reward the presence of cells in the contingency table, which are peaks in both the truth class direction and the cluster direction, as illustrated in Fig. 2. However, the AP metric rewards the presence of such cells more explicitly, since (1) the peakiness metric measures the confidence in the peak cluster population values more effectively as described in the previous paragraphs, and (2) the associativity metric more explicitly and directly rewards the presence of cells that are peaks in both the truth class direction and the cluster direction, as shown in Eqs. 7, 8, and 9.

Thus, the peakiness metric is advantageous over the truth class metric, and the associativity-peakiness metric pair measures associativity more explicitly and directly than the combination of truth class metric and cluster purity from Guo et al.[10]

7. Data Analysis

This section summarizes the findings presented earlier in this report, organized by test scenario, and discusses how the presented data addresses the objectives of the data presentation, namely,

- What does the AP metric tell us that the scikit-learn and F1 metrics do not?

- How well do the scikit-learn metrics and the F1 metric correlate with the AP metric (i.e., how well do they evaluate associativity)?

Furthermore, the following questions are addressed where applicable:

- What about the different metrics accounts for their performance?
- In what situations would the AP metric be most useful and perform better than the scikit-learn metrics?

7.1 Extreme Cases

All metrics under test gave a score of 1.0 to the ideal case, and all but three gave a score of 0 to the worst case. As expected, Completeness gave a score of 1 to that worst case, since in that worst case, the clusters from each truth class were entirely contained within a single cluster—the condition for a score of 1 for the Completeness metric. Thus, Completeness alone is not a suitable metric for comparing clustering algorithms, though it could work well as part of the V-Measure.

The other two metrics that gave a nonzero score to the worst case were FMS and F1. The definitions of these metrics—Eq. 5 for FMS and Eqs. 1 and 2 for F1—show that these metrics would not give a zero score to this worst case. Thus, FMS and F1 would not be effective metrics for evaluating algorithms with very poor performance.

In the following sections on low- and higher-performance contingency tables, the performance of ARS was similar to that of the entropy-based metrics, and is grouped together with them in the performance analysis. However, the performance of ARS does differ from that of the entropy-based metrics in that its dynamic range was slightly higher than that of the entropy-based metrics in all of the test cases. The histograms for each of the test cases (Figs. 8, 15, 18, and 20) showed that ARS had more instances of very low scores, as well as of very high scores, as compared with the entropy-based metrics.

7.2 Low Performance

ARS, as well as the entropy-based metrics, AMI, Completeness, Homogeneity, and V-Measure, gave low scores to the set of contingency tables—the most common value was about 0.15 for all of them. The low scores are not surprising, since the contingency tables in this test scenario were generated randomly, and the entropy-based methods were designed to give low scores to random class label assignments. However, none of these metrics gave a zero score to any of the contingency tables.

The metrics not based on entropy, FMS and F1, gave much higher scores—their values were 0.4 and 0.5, respectively. As the analysis from the extreme cases showed, these metrics are not designed to give low scores to contingency tables that indicate low performance.

Four selected contingency tables from this test case—two with the highest AP score and two with zero AP score—were selected for more detailed analysis. For the contingency tables with highest AP scores, ARS and the entropy-based metrics all gave very low scores (between 0.2 and 0.3) to one contingency table but gave higher scores (between 0.4 and 0.5) to the other contingency table. This discrepancy, as well as the low correlation scores between these metrics and the AP metric, shows that ARS and the entropy-based methods are measuring something different than associativity and peakiness. The metrics that are not entropy-based, FMS and F1, gave higher scores (ranging from 0.4 to 0.7) for these datasets. However, their scores do not come near the scores that the AP metric gave (0.8) to these very high-performing contingency tables.

Regarding the contingency tables with zero AP scores, the performance of all the metrics was similar to their performance with the worst-case contingency table in Section 7.1, "Extreme Cases." The non-entropy-based metrics, FMS and F1, did not give low scores to these contingency tables. As mentioned earlier, these two metrics do not regard the occurrence of all truth-class peaks in the same cluster as an undesirable phenomenon.

During deployment of the clustering algorithm, if one cluster's total population is much larger than all the rest of the cluster totals, which occurs when the AP score is 0, then the user might think that there is only one cluster and thus only one truth class, and discard the remaining clusters as noise. This would be a severe error, since there are actually multiple truth classes, and thus the case of AP score equal to zero is quite a detrimental situation. Therefore, it is very important for a contingency table metric to identify cases in which the data points are all erroneously grouped into one or two clusters (i.e., many fewer clusters than the number of truth classes).

7.3 Higher Performance, 4x4

The score distribution plots in the higher-performance case are generally shifted toward higher scores as compared with the comparable plots for the low-performance 4x4 contingency tables, confirming the designation of higher-performance contingency tables.

None of the metrics showed a very high correlation with the AP metric, though F1 had the highest correlation at 0.602.

7.4 Higher Performance, 4x6

Increasing the number of clusters produced by clustering algorithms from 4 to 6 with four truth classes does not significantly affect the distribution of the scores of any of the metrics tested here.

7.5 Higher Performance, 4x2

The 4x2 case is similar to the worst case of the Extreme Cases section (Section 7.1). In the worst case, all data samples are in one cluster. In the 4x2 case, there are only two clusters, so all data samples are members of one of these two clusters. In both of these cases, there is significant obscuration of the class differences among the data samples. As could be expected based on the worst-case performance, ARS and the entropy-based metrics, AMI, Completeness, Homogeneity, and V-Measure, had many more scores at the low end of the scale, including many zero scores.

For the FMS and F1 metrics, the entire score distributions shifted toward higher values, reach as high a 1 in this test case. These two metrics give high scores in a test case with only two clusters and four truth classes. This is quite a detrimental performance for the FMS and F1 metrics, since they do not recognize the obscuration of truth classes that results from clustering all the data points into just a few clusters, many clusters fewer than the number of truth classes.

Notably, in the 4x2 case, only the AP score gave no scores of 1, whereas all the other metrics did give some scores of 1 or close to 1. This could be explained by the impossibility of getting a high associativity score in contingency matrices with fewer clusters than truth classes. In such cases, it is impossible for there to be a one-to-one matching of clusters with truth classes.

8. Computational Complexity Analysis

An important consideration in evaluating clustering metrics is the execution time of the metric, which depends on the computational complexity of the metric. Big-O formulas for the computational complexity of the metric are not presented here, since these formulas depend on the particular implementation of the metric.

For the various metrics presented here, the execution time was measured for each of 500 instances of computation of the metric. For the AMI, ARS, FMS, Completeness, Homogeneity, and V-Measure metrics, the scikit-learn implementation was used. For the F1 score and AP score, the authors' implementation was used.

The execution time measurement of the scikit-learn implementations included only the time to import the code for the metric and the time to execute the metric. It did not include the time to transform the contingency table to a truth–prediction vector pair.

The computer used to execute the calculations of the metrics is a Dell Precision 7730 laptop. The processor is an Intel Core i7-8850H CPU at 2.60 GHz, with 6 cores and 12 logical processors. The installed physical memory is 32.0 GB.

The measured computation times in milliseconds for each of the metrics for each of the 500 trials are shown in Fig. 23.

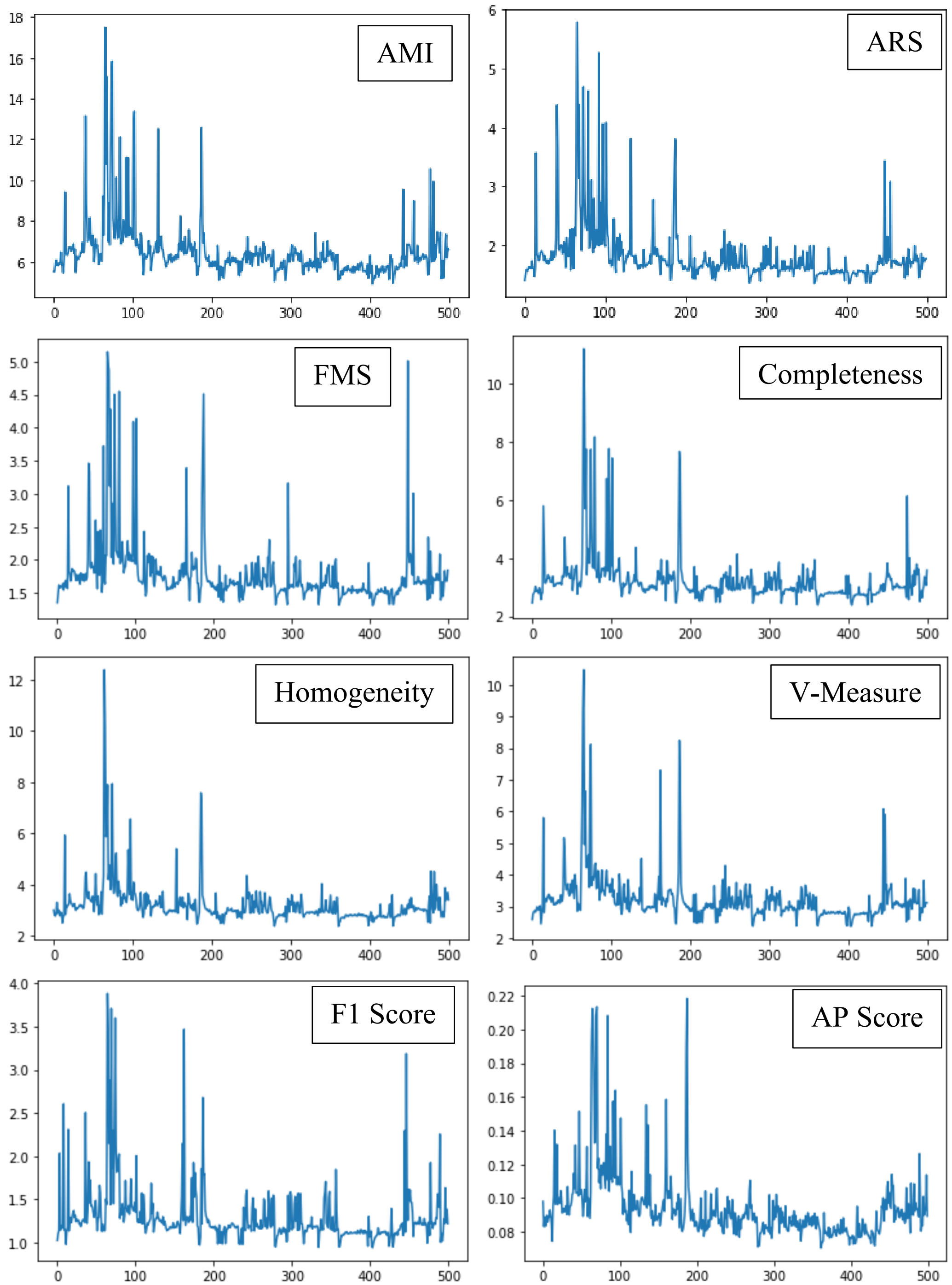

Fig. 23 Execution time plots; x-axis is instance number, and y-axis is time in milliseconds

The plots in Fig. 23 all show distinct patterns of sharp peaks between instances 0 and 200, and between 400 and 500. A likely explanation for these peaks is that they indicate that the computer which was used to calculate these metrics was running some other processes in the background that slowed the execution of the code used to generate the metrics. A relatively quiescent period occurred between indices 200 and 400. Therefore, the mean values for execution time for each metric is calculated only for values whose index is between 200 and 400. The mean execution times for each metric during this interval are shown in Table 12.

Table 12 Table of mean execution times for the metrics

Metric	Mean execution time (ms)
AP	0.085
AMI	5.967
ARS	1.645
FMS	1.629
Completeness	2.985
Homogeneity	2.967
V-Measure	2.991
F1	1.197

Table 12 shows that the entropy-based metrics (AMI, Completeness, Homogeneity, and V-Measure) took the longest, followed by the pairwise-based metrics (ARS and FMS). The fastest metrics were the metrics designed for contingency tables (F1 and AP). Of these two fastest metrics, AP ran in less than 1/10 of the time that it took the F1 metric to run. Thus, for the 200 contingency tables analyzed, the AP metric executed most quickly. Further research to develop analytical expressions for the computational complexity of these metrics would be quite useful, since the data presented in Table 12 is for just one value of N, the number of samples.

9. Conclusions

The main findings in this report are that ARS and the entropy-based metrics gave low scores for the low-performance data, which was randomly generated, as could be expected. However, these metrics tended to give low scores in general, even for the completely nonrandom contingency table of Fig. 1. The non-entropy-based metrics, FMS and F1, did not give zero scores, or even low scores, to even the worst-case test scenario. These two metrics generally gave higher scores than the entropy-based metrics. Thus, ARS and the entropy-based metrics were good at giving low scores to poor performance, and the non-entropy-based metrics were good at giving high scores to good performance, but only the AP metric consistently did both. The AP metric has higher dynamic range, and therefore higher sensitivity, than the scikit-learn metrics and the F1 metric.

The F1 metric was compared with the AP metric. The comparison showed that the AP metric was better at detecting low peakiness in the set of cluster populations for a particular truth class. Also, the AP metric score was not reduced by the presence of several low-population clusters, whereas the F1 metric score was.

The scatter plots and correlation coefficients showed that the scikit-learn metrics and the F1 metric had only moderate correlation with the AP metric, generally about 0.5. The AP metric's execution time was less than that of all the other metrics tested in the experiment described earlier.

10. References

1. scikit-learn developers. 2.3 Clustering – scikit-learn 1.3.0 documentation. Scikit-learn.org; 2023 [accessed 2023 Aug 15]. https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation.

2. Altham PM. The measurement of association of rows and columns for an r×s contingency table. Journal of the Royal Statistical Society. Series B (Methodological). 1970;32(1):63–73.

3. Altham PM. The measurement of association in a contingency table: three extensions of the cross-ratios. Journal of the Royal Statistical Society. Series B (Methodological). 1970;32(3):395–407.

4. Kateri M. Contingency table analysis. Springer; 2014. ISBN 978-0-8176-4811-4 (eBook).

5. Bouchet-Valat M. General marginal-free association indices for contingency tables: from the altham index to the intrinsic association coefficient. Sociological Methods & Research. 2022;51(1):203–236.

6. Silveira PSP, Siqueira JO. Better to be in agreement than in bad company. Behavior Research Methods; 2022. https://doi.org/10.3758/s13428-022-01950-0.

7. Sayal R, Kumar DVV. A novel similarity measure for clustering categorical data sets. International Journal of Computer Applications. 2011;17(1):25–30.

8. Pfitzner D, Leibbrandt R, Powers D. Characterization and evaluation of similarity measures for pairs of clusterings. Knowl Inf Syst. 2009;19:361–394.

9. Dom BE. An information-theoretic external cluster-validity measure. Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI2002); 2012. arXiv:1301.0565 [cs.LG].

10. Guo S, Arif D, Li J. Agile radar signal clustering and platform tracking. NATO SCI-326 Meeting; 2023; Bonn, Germany. (CUI)

11. Manning CD, Raghavan P, Schütze H. Introduction to information retrieval. Cambridge University Press; 2008 July 7. ISBN 978-0-521-86571-5. (Also available in: Cluster analysis. Wikipedia; last updated 2023 Nov 15 [accessed 22 August 2023]. https://en.wikipedia.org/wiki/Cluster_analysis#Evaluation_and_assessment.)

12. scikit-learn developers. 2.3 Clustering–- scikit-learn 1.3.0 documentation; 2.3.11.8 Contingency Matrix. Scikit-learn.org; 2023 [accessed 2023 Aug 17]. https://scikit-learn.org/stable/modules/clustering.html#contingency-matrix.

13. scikit-learn developers. 2.3 Clustering–- scikit-learn 1.3.0 documentation; 2.3.11.1 Rand index. Scikit-learn.org; 2023 [accessed 2023 Aug 17]. https://scikit-learn.org/stable/modules/clustering.html#rand-index.

14. datasciencewithchris.com. Evaluate clustering algorithms; n.d. [accessed 2023 Aug 17]. https://datasciencewithchris.com/evaluate-clustering-algorithms/.

15. Vinh NX, Epps J, Bailey J. Information theoretic measures for clusterings comparison. Proceedings of the 26th Annual International Conference on Machine Learning – ICML '09; 2009. p. 1. doi:10.1145/1553374.1553511. ISBN 9781605585161. (Also available in: Adjusted mutual information. Wikipedia; last updated 2022 Dec 23 [accessed 2023 Aug 17]. https://en.wikipedia.org/wiki/Adjusted_mutual_information.)

16. scikit-learn developers. 2.3 Clustering - scikit-learn 1.3.0 documentation; 2.3.11.4 Fowlkes-Mallows scores. scikit-learn.org; 2023 [accessed 2023 Aug 17]. https://scikit-learn.org/stable/modules/clustering.html#fowlkes-mallows-scores.

17. Rosesnberg A, Hirschberg J. V-Measure: a conditional entropy-based external cluster evaluation measure. Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning; 2007; Prague.

18. Dickinson M. Python - generate random numbers summing to a predefined value - stack overflow. 2010 Aug 28 [accessed 2023 Aug 22]. https://stackoverflow.com/questions/3589214/generate-random-numbers-summing-to-a-predefined-value.

19. docs.python.org. Random - generate pseudo-random numbers - Python 3.11.4. documentation; 2023 Aug 22 [accessed 2023 Aug 22]. https://docs.python.org/3/library/random.html#random.choice.

20. Bobbitt Z. What is considered to be a "strong" correlation? – statology. Statology.org; 2020 Jan 22 [accessed 2023 Aug 24]. https://www.statology.org/what-is-a-strong-correlation/.

Appendix. Demonstration of Equivalency of Contingency Table and Truth–Prediction Vector Pair

This Appendix demonstrates the equivalence between truth–prediction vector pairs and contingency tables. It shows the procedure for transformation from one domain to the other. This Appendix also shows that the mapping from truth–prediction vector pairs to contingency tables is many-to-one and states the constraint for a set of truth–prediction vector pairs to be transformed to the same contingency table.

This equivalence between truth–prediction vector pairs and contingency tables enables application of metrics for truth–prediction vector pairs to be applied to contingency tables and vice versa.

A.1 Derivation of Contingency Table from Truth–Prediction Vector Pair

This section shows how a contingency table can be derived from a truth–prediction vector pair.

Define a vector pair of truth elements T and their corresponding prediction elements P, each with N elements, as shown in Eqs. A1 and A-2.

$$Truth\ vector\ T[i], i = 0\ to\ N - 1 \tag{A-1}$$

$$Prediction\ vector\ P[i], i = 0\ to\ N - 1 \tag{A-2}$$

Here, N is the number elements of both T and P and represents the number of data samples used in the clustering process.

From the truth vector T and the prediction vector P, we can derive two new vectors $Tvals$ and $Cinds$, which are the set of unique truth values and the set of unique cluster indices, respectively, as shown in Eqs. A-3 and A-4.

$$Tvals = unique(T) \tag{A-3}$$

$$Cinds = unique(P) \tag{A-4}$$

Here, the function $unique$ selects the set of unique values in the set that is its argument. Various software packages, including the NumPy library of Python,[1] implement this function.

At this point, we can define the elements of the contingency table CT as shown in Eq. A-5.

$$contingnency\ table\ CT[m, n], for\ m = 0\ to\ NT - 1, and\ n = 0\ to\ NC - 1 \tag{A-5}$$

[1] NumPy developers. numpy.unique - numPy v1.25 manual; 2022 [accessed 2023 Aug 29]. https://numpy.org/doc/stable/reference/generated/numpy.unique.html.

In contingency table CT, the rows are truth classes and the columns are clusters. The elements of CT are derived from the elements of the truth vector T and the prediction vector P as shown in Eq. A-6.

$$CT[m,n] = number\ of\ indices\ i\ for\ which\ T[i] = Tvals[m]\ and\ P[i] = Cinds[n] \quad \text{(A-6)}$$

In Eq. A-6, the variable i ranges from 0 to $N-1$, as shown in Eqs. A-1 and A-2.

Even if the truth vector T and the prediction vector P are reordered in some manner, as long as they are both reordered in the same manner such that $T[k]$ and $P[k]$ continue to have the same index as each other for all k from 0 to $N-1$, the resulting contingency table will be the same. Thus, the mapping of truth–prediction vector pairs to contingency tables is a many-to-one mapping.

A.2 Derivation of Truth–Vector Prediction Pair from Contingency Table

This section shows how a truth–prediction vector pair can be derived from a contingency table.

Define a contingency table CT as shown in Eq. A-5. As in Eq. A-5, the rows of CT are truth classes and columns of CT are clusters. From the contingency table CT, we can derive the vector of truth classes, $Tvals$, which has NT elements, and the vector of cluster indices $Cinds$, which has NC elements, as shown in Eqs. A-7 and A-8.

$$Vector\ of\ truth\ values\ Tvals[m]\ , for\ m = 0\ to\ NT-1 \quad \text{(A-7)}$$

$$Vector\ of\ cluster\ indices\ Cinds[n], for\ n = 0\ to\ NC-1 \quad \text{(A-8)}$$

At this point, we can derive a truth–vector prediction pair consisting of a truth vector T and a prediction vector P by iterating through all the elements of the contingency table CT as shown in Eqs. A-9, A-10, and A-11.

$$For\ each\ m,n\ for\ which\ m = 0\ to\ NT-1, and\ n = 0\ to\ NC-1: \quad \text{(A-9)}$$

$$Create\ CT[m,n]\ elements\ of\ T\ with\ value\ Tvals[m] \quad \text{(A-10)}$$

$$Create\ CT[m,n]\ elements\ of\ P\ with\ value\ Cinds[n] \quad \text{(A-11)}$$

The truth–prediction vector pair that is derived from contingency table CT could have various ordering of elements, depending on the order of iteration through the indices m,n for example. All orderings of the elements of the truth vector and prediction vector will give rise to the same contingency table using the procedure in Eqs. A-4 through A-6, if and only if the condition in Eq. A-12 is satisfied.

$$\text{The number of rows of } T \text{ and } P \text{ such that } [T[i] = Tvals[m] \text{ and } P[i] = Cinds[n]\ ] = CT[m,n] \tag{A-12}$$

In Eq. A-12, the index i ranges from 0 to $N - 1$, where N is the number of elements of both the truth vector T and the prediction vector P, and is equal to the sum total of all of the elements in the contingency table CT.

A single contingency table can give rise to truth–prediction vector pairs with many different element orderings, such that each element ordering satisfies the constraint in Eq. A-12. Thus, the mapping of contingency tables to truth–prediction vector pairs is one-to-many.

List of Symbols, Abbreviations, and Acronyms

2D	two-dimensional
AMI	Adjusted Mutual Information
AP	Associativity–Peakiness
ARS	Adjusted Rand Score
FM	Fowlkes–Mallows
FMS	Fowlkes–Mallows Score
FN	false negative
FP	false positive
MI	Mutual Information
TP	true positive

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DEVCOM ARL FCDD RLB CI TECH LIB
2 (PDF)	DEVCOM ARL FCDD RLA LD N ZIRKIND W DIEHL