MT4UML: Metamorphic Testing for Unsupervised Machine Learning

Faqeer ur Rehman &

Dr. Clemente Izurieta

Outline

- Introduction
- Challenges
- Background
- Existing Work and Limitation
- Proposed Approach
- Results
- Conclusion
- Future Work

Machine Learning (ML) Testing

- Machine Learning (ML) applications are becoming very popular in different domains e.g., healthcare, finance, cyber-security, self-driving cars, and aircraft collision avoidance systems.

- A small bug in the system may lead to catastrophic failure which can result in both financial and human loss.

- Due to their wide adoptability in society, it is essential to perform testing of ML applications from multiple perspectives.

Challenges

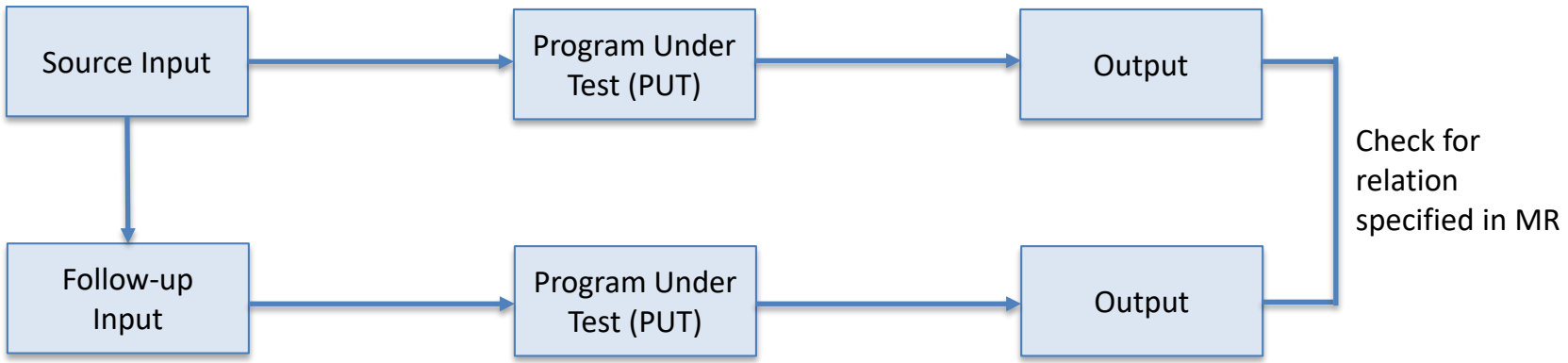
Testing ML applications poses several challenges, which are as following.

- ML applications have a large input space for which they need to be verified.
- The low accuracy of ML model is a composite effect, which can arise from a combination of three ML components namely: data, program and the framework/library.
- Testing ML applications seriously suffer from the Oracle problem.

Background: Metamorphic Testing (MT)

- Metamorphic Testing uses Metamorphic Relations (MRs) to find defects in the program.

- MRs are the necessary properties of a program under test that specifies how the output should be changed on changing the related input.



K-means Clustering

$$c_i^{(t+1)} = \frac{1}{|C_i^{(t)}|} \sum_{j=1}^{n} x_j$$

- $c_i^{(t+1)}$ represents the i^{th} new centroid found, and x_j represents the j^{th} instance (where j = 1, 2, ..., n) belonging to the cluster C_i.

Agglomerative Clustering

- Average Linkage Method.

$$d(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x_r \in C_i} \sum_{x_s \in C_j} d(x_r, x_s)$$

- $d(C_i , C_j)$ represents the distance between cluster C_i and cluster C_j , x_r represents the r^{th} data point (where r = 1,2,...,n) belonging to the cluster C_i , and x_s represents the s^{th} data point (where s = 1,2,...,n) belonging to the cluster C_j.

Existing Solutions in Literature

- To the best of our knowledge, we are able to find only two research papers in which MT has been utilized to test unsupervised clustering algorithms (provided by WEKA tool) [1] [2].

- This research helps end-users (non-technical users) coming from diverse fields such as bioinformatics, finance, and electrical engineering to choose a specific type of algorithm from a large set of available algorithms that can best fit their needs.

Limitations

However, the following are the limitations we have found in their work:

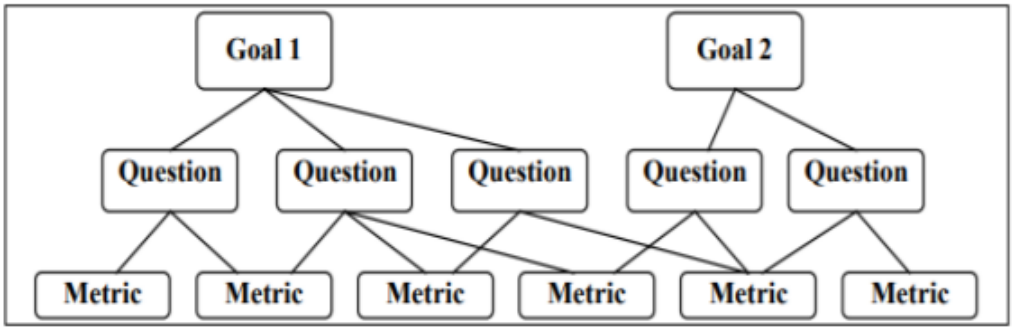
1. The proposed approaches only serve validation purposes, and only check whether the algorithms under test meet the user expectations.

2. The proposed MRs only target the algorithms provided by the WEKA tool. It is equally worth exploring to test the behavior of other notable clustering algorithms provided by widely used Python libraries i.e., Scikit-learn.

3. The proposed approaches use synthetic 2D data (i.e., not real data). It is worth exploring whether the proposed MRs are effective in testing the models that use multidimensional real-world data sets as well.

Proposed Solution

- The GQM (Goal Question Metric) [3] is a goal-oriented approach that we have used to frame the research work.



GQM

- **Research Goal (RG):** To investigate the MT technique for testing unsupervised algorithms *for the purpose of* improving their quality *from the perspective of* both the end user and a developer *in the context of* testing K-means and Agglomerative clustering algorithms.

- **Research Question 1 (RQ1):** How effective are the proposed MRs in testing the clustering algorithms under test?
- **Research Question 2 (RQ2):** Which algorithm is more stable for performing clustering-related tasks?

- **Research Metric 1 (RM1):** Number of violated MRs - A count of the number of MRs that are violated by the algorithms under test.
- **Research Metric 2 (RM2):** Violation Rate - Percentage of instances for which the programs show inconsistent behavior.

Contributions

- We propose an MT based approach for verification and validation of two popular clustering algorithms, provided by the leading Python library known as scikit-learn.
- We propose 22 MRs to assess the behavior (from both the user's and developer's/implementation perspective) of the clustering algorithms under test.
- The proposed MRs are further analyzed, necessary reasoning is provided, and MRs are then categorized to show whether each of those MRs targets the verification or the validation aspect of testing the two algorithms under investigation.
- The effectiveness of the proposed approach is demonstrated by applying it to testing an open-source customer segmentation application.

Metamorphic Relations

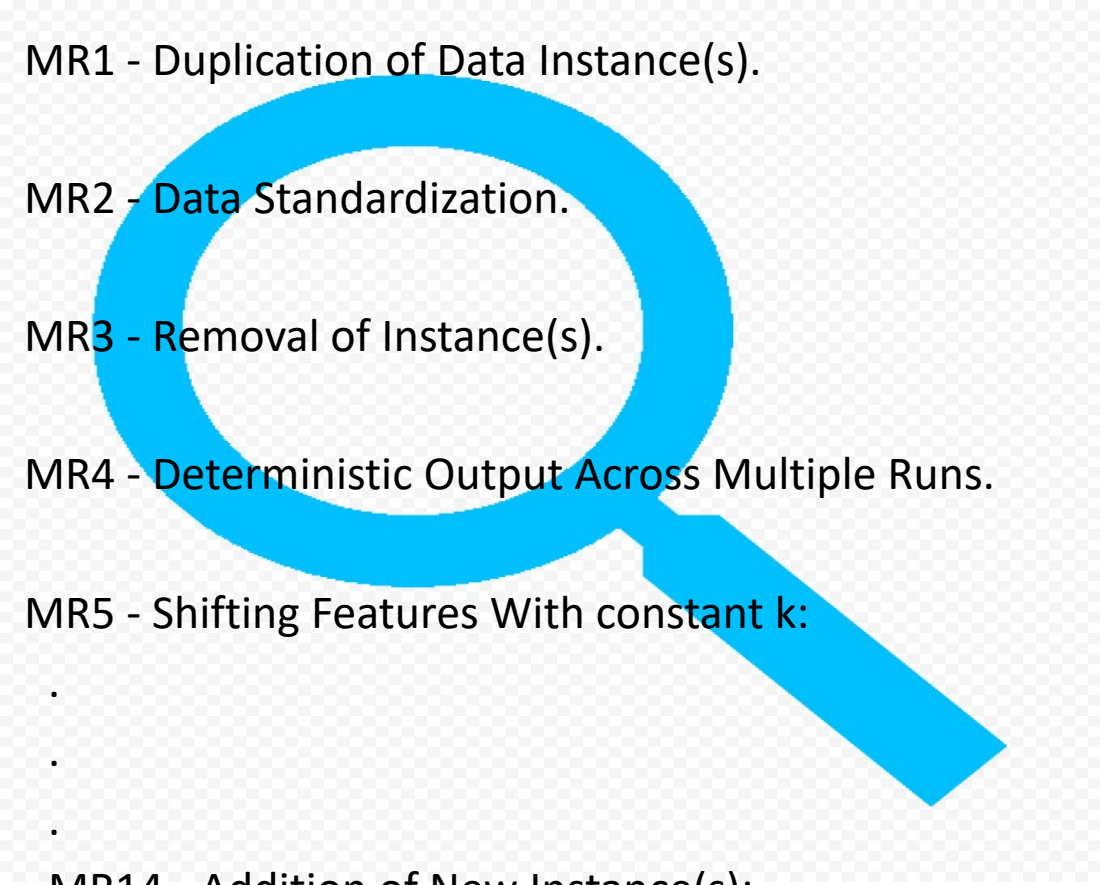
- MR1 - Duplication of Data Instance(s).
- MR2 - Data Standardization.
- MR3 - Removal of Instance(s).
- MR4 - Deterministic Output Across Multiple Runs.
- MR5 - Shifting Features With constant k:

  .

  .

  .
- MR14 - Addition of New Instance(s):

Metamorphic Relations



- MR1 - Duplication of Data Instance(s).
- MR2 - Data Standardization.
- MR3 - Removal of Instance(s).
- MR4 - Deterministic Output Across Multiple Runs.
- MR5 - Shifting Features With constant k:

  .

  .

  .
- MR14 - Addition of New Instance(s):

Metamorphic Relations

- **MR1.1 - Duplication of single instance**: For a given source input s, with associated data instances assigned to clusters c_i (where i = 1, 2, 3, ..., n), we denote the output as O_s. If we duplicate a single instance in the follow-up input f, the output O_f should remain consistent i.e., $O_s = O_f$

- **MR2 - Data Standardization:** If the existing standardized data is once again standardized, the output for both the source and follow-up inputs should remain the same i.e., $O_s = O_f$

Verification & Validation

- The MRs targeting the **verification** aspect aim to check whether the algorithms under test adhere to the specification (from the implementation perspective) expected from the algorithms, whereas,

- The MRs targeting the **validation** aspect aim to check whether the algorithms under test meet the general user expectations or not.

Verification & Validation Analysis (K-Means)

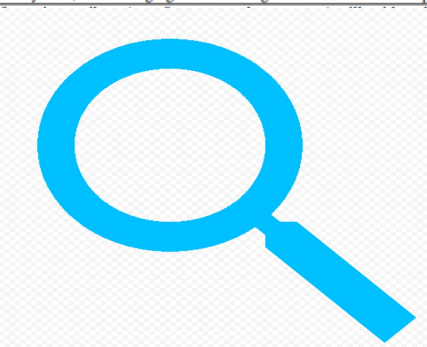
#	VR?	VD?	Reasoning
MR1	×	✓	For the follow-up input, if we add a duplicate instance(s) to any of the clusters, this will result in different cluster centroid(s) (calculated using Equation 1) which may cause the original data points to get assigned to different clusters; thus, changing the output for the follow-up input. It is important to note that this is how (as shown in Equation 1) the centroid/mean calculation is implemented in the K-means algorithm under test, which ultimately means that the violation of this MR can not be characterized as violating the necessary characteristics (related to implementation) of the algorithm under test. Therefore, this MR can not be used for **verification** purposes (because its violation can not be characterized as the bug in the implementation) but can be used for **validation** purposes (because this is what the user's general expectation would be from this algorithm). ***Note**: Readers can use the same reasoning for the verification and validation aspect of testing the algorithms under test for rest of the proposed MRs.*
MR2	✓	✓	For the follow-up input, if we re-apply the standardization step, i) it will not change the mean and variance of the data points, and ii) it will maintain the same distance among the data points (similar to the source-input); thus, it should not not change the results.
MR3	×	✓	For the follow-up intput, if we add a duplicate attribute(s) to the original data points, they may have a strong influence on changing the distance between the data points and their existing centroids; thus, assigning the data points to different clusters. An example is provided (inside excel sheet available in GitHub repo), where, a violation of this MR can be seen.
MR4	×	✓	For the follow-up input, if we remove any instance(s), Equation 1 will result in the calculation of different centroids which ultimately may lead to changing the final output i.e., data points assigned to different clusters. Since this violation can not be characterized to the wrong implementation, it can only be used for validation purposes.
MR5	✓	✓	The addition of uninformative attributes (e.g., 0 or some other constant) will not change the existing relationship between the data points and will also maintain the same relationship between the data points and the initial centroids. Therefore, the output for the follow-up input should remain the same.
MR6	✓	✓	Running the algorithm at different times (keeping the initial centroids the same) should not have any effect on how the centroids are calculated. Further, as there is no change made in the data points, the output should remain the same. If the output changes, it means that there is some implementation bug in the algorithm under test.
MR7	✓	✓	If we shift all the features with some constant k, it will maintain the same distance between the data instances. So, Equation 1 will produce the same centroids as the one found during source execution; thus, not changing the final output. Let x be the centroid, y be the data point and the distance $d(x, y)$ found between them during source execution is z. Now, if we shift the features of both x and y with some constant (i.e., k) then the distance between them would be $d(x, y) = \sqrt{((x + c) - (y + c))^2} => \sqrt{(x - y)^2} => x - y => z$ (i.e., will remain the same). Therefore, the output for both the source and follow-up inputs should remain consistent.
MR8	✓	✓	If we scale all the features with some constant k, it will maintain the same relationship between the data points and the centroids. Let x and y be the two data points and during the source execution their relationship to the centroid c is $x - c < y - c$. During the follow-up execution, after scaling the features with constant k i.e., $k(x - c) < k(y - c) => x - c < y - c$, the relationship (i.e., greater than, less than, and equal) remains the same. Therefore, if the output for the follow-up input changes, that would suggest that there is some bug in the algorithm under test.
MR9	×	✓	If we replace any of the instances with some other instance (belonging to the same cluster), it may result in the calculation of different centroids (as per Equation 1); thus an instance x_i (where $i = 1, 2, .., n$) may get assigned to a different cluster.
MR10	✓	✓	For the follow-up input, if we change the location of features, it will not have any affect on the relationship between the data points and calculation of centroids (using Equation 1). So, the output should remain the same.
MR11	×	✓	For the follow-up input, if we add an informative attribute to each of the clusters instances, it may result in changing the centroids that can assign the instances to different clusters; thus, leading to a different final output.
MR12	✓	✓	For the follow-up input, if we change the order of rows/data points, it will not have any affect on existing relationship between the data point s and will lead to the calculation of same centroids (similar to the one found during source execution). Thus, the output for the source and follow-up inputs should remain consistent.
MR13	✓	✓	For the follow-input, if we apply reflection transformation, the distance between the data points will remain the same thus leading to the calculation of same centroids (using Equation 1). This should result in consistent output for both the source and follow-up inputs.
MR14	×	✓	If we add a new instance(s) to any of the clusters, this addition of new instance(s) may result in the change of centroids (different from the one found during source execution); thus, changing the final output.

Verification & Validation Analysis (Agglomerative Clustering)

#	VR?	VD?	Reasoning
MR1	×	✓	For the follow-up input, if we add a duplicate instance(s) to any of the clusters, this may change the average distance between the two clusters (calculated using Equation 2), thus ending up with changing the clusters for the original data points. For understanding purposes, suppose in Fig. 2, the dendogram for the source input is cut to obtain the two clusters. The data points 7,6,6,10 will be assigned to one cluster, whereas, the data points 2,3 will be assigned to a second cluster. Now, if we duplicate the data point 3 and cut the dendogram to obtain the two clusters, it can be seen in Fig. 3 that the original data points 7,6,6,2,3 are now assigned to one cluster, whereas, the data point 10 is assigned to the second cluster; thus, changing the output for the follow-up inputs.
MR2	✓	✓	Same reasoning as provided for MR2 in Table I
MR3	×	✓	For the follow-up intput, if we add a duplicate attribute(s) to the original data points, they may have a strong influence on changing the average distance between the clusters; thus, changing the overall result. An example is provided (inside excel sheet available in GitHub repo), where, a violation of this MR can be seen.
MR4	×	✓	For the follow-up input, if we remove an instance(s), Equation 2 will result in changing the average distance between the clusters. As an example, in Fig. 2, if we remove the data point '2', the data point '3' (instead of data point '10') will get assigned to the cluster '7,6,6' . Now, if the dendogram is cut to obtain the two clusters, one cluster will have data points '7,6,6,3', whereas, the second one will have only '10', thus changing the clustering result for the follow-up inputs.
MR5	✓	✓	The addition of uninformative attribute (e.g., 0 or some other constant) will neither change the existing relationship between the data points nor the average distance between the clusters. Therefore, the output for the follow-up input should remain the same.
MR6	✓	✓	Running the algorithm at different times should not have any affect on how the average distance between the clusters is calculated. Apart from that, as there is no change made in the data points, so the output should remain consistent.
MR7	✓	✓	If we shift all the features with some constant k, it will maintain the same distance between all the data instances. So, Equation 2 will result in merging the same clusters that were merged during the source execution. As an example, let x and y be the two data-points merged together during the source execution. Now, if we shift the features of both the x and y with some constant 'c' then the distance between them i.e., $d(x,y) = \sqrt{((x+c)-(y+c))^2} => \sqrt{(x-y)^2}$, would remain similar to the one calculated during the source execution. Therefore, the output for both the source and follow-up inputs should remain the same.
MR8	✓	✓	If we scale all the features with some constant k, it will maintain the same relationship (i.e., greater than, less than, and equal) between the data points. As an example, let suppose that we have three data points x, y, and z that we are interested to group them into two clusters. During the source execution, let x and y are merged together to form one cluster, and z in another cluster. The current relationship between them is $x - y < z - y$. During the follow-up execution, after scaling the features with a constant k i.e., $k(x-y) < k(z-y) => x - y < y - z$, the relationship remains the same. Therefore, the output should remain consistent for both the source and follow-up executions.
MR9	×	✓	For the follow-up input, if we replace any of the instances with some other instance (that belongs to the same cluster), it may change the average distance between them (calculated using Equation 2) which can result in assigning the original input to different clusters. As an example, as shown in Fig. 2, if in cluster#2 (which contain data points 2 and 3), we replace the instance 2 with instance 3, it will assign them to the cluster#1 which contains the data points 7,6,6. Now if the dendogram is cut to obtain the two clusters, one cluster will have the data-points 7,6,6,3,3, whereas, the other will have only 10; thus, the original data-point (which is 3) has been assigned to the cluster#1 (instead of cluster#2). This will result in violation of this MR.
MR10	✓	✓	For the follow-up input, if we change the location of features, it will not have any affect on the relationship between the data points and calculation of average distance between the clusters (calculated using Equation 1). So, the output should remain unchanged for both the source and follow-up executions.
MR11	✓	✓	If we add an informative attribute such that it is strongly associated with each of the clusters, it will not change the existing relationship between the data points assigned to each clusters. As an example, let suppose that the dendogram for the source execution (as shown in Fig. 2) is cut to form two clusters, one cluster will have the data points 7,6,6,10, whereas, the other cluster will have the data points 2,3. Now, for the follow-up execution, if we add a new informative attribute which has the value 3 for all the instances in cluster#1 i.e., (7,3),(6,3),(6,3),(10,3) and value 4 for all the instances in cluster#2 i.e., (2,4),(3,4), it will not change the existing relationship between the clusters; thus, the output should remain the same.
MR12	✓	✓	For the follow-up input, if we change the order of rows/data instances, it will not have any affect on the way the calculation is made (using Equation 2) to merge the two clusters. Thus, the output for both the source and follow-up inputs should remain the same.
MR13	✓	✓	For the follow-input, if we apply reflection transformation to all data points, the distance between them will remain the same; thus, leading to the identification of the same clusters found during source execution.
MR14	×	✓	If we add a new instance(s) to any of the clusters, this addition of new instance(s) may result in change of average distance between the clusters; thus, changing the final output. An example is provided (inside excel sheet available in GitHub repo), where, a violation of this MR can be seen.

Verification & Validation Analysis (Agglomerative Clustering)

#	VR?	VD?	Reasoning
MR1	×	✓	For the follow-up input, if we add a duplicate instance(s) to any of the clusters, this may change the average distance between the two clusters (calculated using Equation 2), thus ending up with changing the clusters for the original data points. For understanding purposes, suppose in Fig. 2, the dendogram for the source input is cut to obtain the two clusters. The data points 7,6,6,10 will be assigned to one cluster, whereas, the data points 2,3 will be assigned to a second cluster. Now, if we duplicate the data point 3 and cut the dendogram to obtain the two clusters, it can be seen in Fig. 3 that the original data points 7,6,6,2,3 are now assigned to one cluster, whereas, the data point 10 is assigned to the second cluster; thus, changing the output for the follow-up inputs.
MR2	✓	✓	Same reasoning as provided for MR2 in Table I
MR3	×	✓	For the follow-up intput, if we add a duplicate attribute(s) to the original data points, they may have a strong influence on changing the average distance between the clusters; thus, changing the overall result. An example is provided (inside excel sheet available in GitHub repo), where, a violation of this MR can be seen.
MR4	×	✓	For the follow-up input, if we remove an instance(s), Equation 2 will result in changing the average distance between the clusters. As an example, in Fig. 2, if we remove the data point '2', the data point '3' (instead of data point '10') will get assigned to the cluster '7,6,6' . Now, if the dendogram is cut to obtain the two clusters, one cluster will have data points '7,6,6,3', whereas, the second one will have only '10', thus changing the clustering result for the follow-up inputs.
MR5	✓	✓	The addition of unin… e the existing relationship between the data points nor the avera… /-up input should remain the same.
MR6	✓	✓	Running the algorith… distance between the clusters is calculated. Apart from that, as … ain consistent.
MR7	✓	✓	If we shift all the f… tween all the data instances. So, Equation 2 will result in mer… n. As an example, let x and y be the two data-points merged … of both the x and y with some constant 'c' then the distanc… $-y)^2$, would remain similar to the one calculated during th… low-up inputs should remain the same.
MR8	✓	✓	If we scale all the … ip (i.e., greater than, less than, and equal) between the data poi… and z that we are interested to group them into two clusters. D… one cluster, and z in another cluster. The current relationship … fter scaling the features with a constant k i.e., $k(x-y) < k(…$ refore, the output should remain consistent for both the source …
MR9	×	✓	For the follow-up in… belongs to the same cluster), it may change the average distance … ning the original input to different clusters. As an example, as s… ve replace the instance 2 with instance 3, it will assign them to … gram is cut to obtain the two clusters, one cluster will have the … e original data-point (which is 3) has been assigned to the cluster#1 (instead of cluster#2). This will result in violation of this MR.
MR10	✓	✓	For the follow-up input, if we change the location of features, it will not have any affect on the relationship between the data points and calculation of average distance between the clusters (calculated using Equation 1). So, the output should remain unchanged for both the source and follow-up executions.
MR11	✓	✓	If we add an informative attribute such that it is strongly associated with each of the clusters, it will not change the existing relationship between the data points assigned to each clusters. As an example, let suppose that the dendogram for the source execution (as shown in Fig. 2) is cut to form two clusters, one cluster will have the data points 7,6,6,10, whereas, the other cluster will have the data points 2,3. Now, for the follow-up execution, if we add a new informative attribute which has the value 3 for all the instances in cluster#1 i.e., (7,3),(6,3),(6,3),(10,3) and value 4 for all the instances in cluster#2 i.e., (2,4),(3,4), it will not change the existing relationship between the clusters; thus, the output should remain the same.
MR12	✓	✓	For the follow-up input, if we change the order of rows/data instances, it will not have any affect on the way the calculation is made (using Equation 2) to merge the two clusters. Thus, the output for both the source and follow-up inputs should remain the same.
MR13	✓	✓	For the follow-input, if we apply reflection transformation to all data points, the distance between them will remain the same; thus, leading to the identification of the same clusters found during source execution.
MR14	×	✓	If we add a new instance(s) to any of the clusters, this addition of new instance(s) may result in change of average distance between the clusters; thus, changing the final output. An example is provided (inside excel sheet available in GitHub repo), where, a violation of this MR can be seen.



Verification & Validation Analysis

- **MR-1: Duplication of Instance(s)**

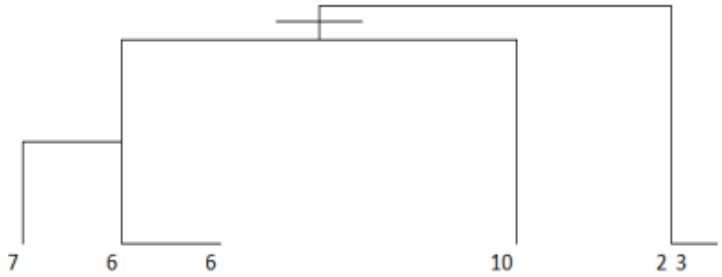
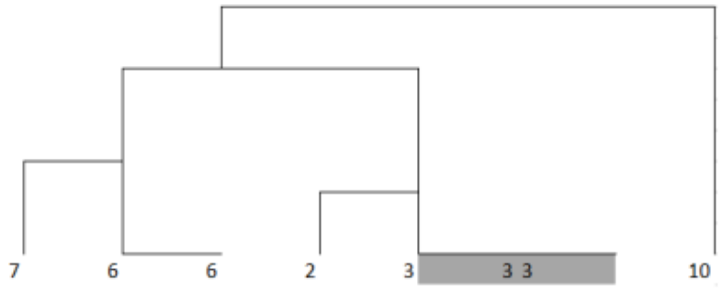


Fig. 2. Agglomerative Clustering Example



3. MR1 for agglomerative clustering: Added 3 as a duplicate instance

Verification & Validation Analysis

- **MR-2 - Data Standardization:**

  For the follow-up input, if we re-apply the standardization step, i) it will not change the mean and variance of the data points, and ii) it will maintain the same distance among the data points (similar to the source-input); thus, it should not change the results.

Results

RESULTS OF TESTING K-MEANS AND AGGLOMERATIVE CLUSTERING ALGORITHMS

	K-Means				Agglomerative Clustering	
MR#	Same cluster assigned?	Centroids same?	Nearest point to centroid(s) same?	Violation? (Violation rate)	Same cluster assigned?	Violation? (Violation rate)
1.1	×	×	✓	✓ (0.12%)	×	✓ (0.05%)
1.2	×	×	✓	✓ (0.10%)	×	✓ (98.53%)
1.3	✓	✓	✓	×	N/A	N/A
2	✓	✓	✓	×	✓	×
3	✓	✓	✓	×	✓	×
4.1	×	×	✓	✓ (0.14%)	×	✓ (0.09%)
4.2	×	×	✓	✓ (0.23%)	×	✓ (0.23%)
4.3	×	×	×	✓ (57.82%)	×	✓ (93.40%)
5	✓	✓	✓	×	✓	×
6	✓	✓	✓	×	✓	×
7	✓	✓	✓	×	✓	×
8	✓	✓	✓	×	✓	×
9.1	✓	×	✓	✓	×	✓ (2.14%)
9.2	✓	✓	✓	×	×	✓ (98.07%)
9.3	×	×	✓	✓ (0.02%)	×	✓ (99.28%)
10	✓	✓	✓	×	✓	×
11	✓	×	×	✓	✓	×
12.1	✓	✓	✓	×	✓	×
12.2	✓	✓	✓	×	✓	×
13	✓	✓	✓	×	✓	×
14.1	×	×	✓	✓ (0.05%)	×	✓ (0.28)
14.2	✓	×	×	✓	×	✓ (94.90%)

Results

RESULTS OF TESTING K-MEANS AND AGGLOMERATIVE CLUSTERING ALGORITHMS

	K-Means				Agglomerative Clustering	
MR#	Same cluster assigned?	Centroids same?	Nearest point to centroid(s) same?	Violation? (Violation rate)	Same cluster assigned?	Violation? (Violation rate)
1.1	×	×	✓	✓(0.12%)	×	✓(0.05%)
1.2	×	×	✓	✓(0.10%)	×	✓(98.53%)
1.3	✓	✓	✓	×	N/A	N/A
2	✓	✓	✓	×	✓	×
3	✓	✓	✓	×	✓	×
4.1	×	×	✓	✓(0.14%)	×	✓(0.09%)
4.2	×	×	✓	✓(0.23%)	×	✓(0.23%)
4.3	×	×	×	✓(57.82%)	×	✓(93.40%)
5	✓	✓	✓	×	✓	×
6	✓	✓	✓	×	✓	×
7	✓	✓	✓	×	✓	×
8	✓	✓	✓	×	✓	×
9.1	✓	×	✓	✓	×	✓(2.14%)
9.2	✓	✓	✓	×	×	✓(98.07%)
9.3	×	×	✓	✓(0.02%)	×	✓(99.28%)
10	✓	✓	✓	×	✓	×
11	✓	×	×	✓	✓	×
12.1	✓	✓	✓	×	✓	×
12.2	✓	✓	✓	×	✓	×
13	✓	✓	✓	×	✓	×
14.1	×	×	✓	✓(0.05%)	×	✓(0.28)
14.2	✓	×	×	✓	×	✓(94.90%)

Results

RESULTS OF TESTING K-MEANS AND AGGLOMERATIVE CLUSTERING ALGORITHMS

	K-Means				Agglomerative Clustering	
MR#	Same cluster assigned?	Centroids same?	Nearest point to centroid(s) same?	Violation? (Violation rate)	Same cluster assigned?	Violation? (Violation rate)
1.1	×	×	✓	✓ (0.12%)	×	✓ (0.05%)
1.2	×	×	✓	✓ (0.10%)	×	✓ (98.53%)
1.3	✓	✓	✓	×	N/A	N/A
2	✓	✓	✓	×	✓	×
3	✓	✓	✓	×	✓	×
4.1	×	×	✓	✓ (0.14%)	×	✓ (0.09%)
4.2	×	×	✓	✓ (0.23%)	×	✓ (0.23%)
4.3	×	×	×	✓ (57.82%)	×	✓ (93.40%)
5	✓	✓	✓	×	✓	×
6	✓	✓	✓	×	✓	×
7	✓	✓	✓	×	✓	×
8	✓	✓	✓	×	✓	×
9.1	✓	×	✓	✓	×	✓ (2.14%)
9.2	✓	✓	✓	×	×	✓ (98.07%)
9.3	×	×	✓	✓ (0.02%)	×	✓ (99.28%)
10	✓	✓	✓	×	✓	×
11	✓	×	×	✓	✓	×
12.1	✓	✓	✓	×	✓	×
12.2	✓	✓	✓	×	✓	×
13	✓	✓	✓	×	✓	×
14.1	×	×	✓	✓ (0.05%)	×	✓ (0.28)
14.2	✓	×	×	✓	×	✓ (94.90%)

Results

RESULTS OF TESTING K-MEANS AND AGGLOMERATIVE CLUSTERING ALGORITHMS

	K-Means				Agglomerative Clustering	
MR#	Same cluster assigned?	Centroids same?	Nearest point to centroid(s) same?	Violation? (Violation rate)	Same cluster assigned?	Violation? (Violation rate)
1.1	×	×	✓	✓(0.12%)	×	✓(0.05%)
1.2	×	×	✓	✓(0.10%)	×	✓(98.53%)
1.3	✓	✓	✓	×	N/A	N/A
2	✓	✓	✓	×	✓	×
3	✓	✓	✓	×	✓	×
4.1	×	×	✓	✓(0.14%)	×	✓(0.09%)
4.2	×	×	✓	✓(0.23%)	×	✓(0.23%)
4.3	×	×	×	✓(57.82%)	×	✓(93.40%)
5	✓	✓	✓	×	✓	×
6	✓	✓	✓	×	✓	×
7	✓	✓	✓	×	✓	×
8	✓	✓	✓	×	✓	×
9.1	✓	×	✓	✓	×	✓(2.14%)
9.2	✓	✓	✓	×	×	✓(98.07%)
9.3	×	×	✓	✓(0.02%)	×	✓(99.28%)
10	✓	✓	✓	×	✓	×
11	✓	×	×	✓	✓	×
12.1	✓	✓	✓	×	✓	×
12.2	✓	✓	✓	×	✓	×
13	✓	✓	✓	×	✓	×
14.1	×	×	✓	✓(0.05%)	×	✓(0.28)
14.2	✓	×	×	✓	×	✓(94.90%)

Conclusion

- First, we propose Metamorphic Testing (MT) based approach for performing better quality assurance of clustering algorithms.

- Second, we propose a broader set of 22 MRs that both researchers and practitioners can take advantage of to assess the behavior of the clustering algorithms under test from both the user's general expectation (validation) and from the implementation perspective (verification).

Conclusion

- For testing the K-means algorithm, we also propose multiple criteria that can be used for verification of the MRs. This also opens a new research direction for researchers to make fruitful contributions.

- Our results show that both the algorithms under test exhibit violations (from the validation perspective) for 10 MRs. Further, in comparison to K-means, the Agglomerative clustering algorithm is highly susceptible to small changes in inputs and may not offer a better alternative to scenarios captured by the violated MRs.

Future work

- To improve the testing of agglomerative clustering-based applications, we intend to develop new criteria (similar to testing k-means) that can be used to verify MRs.

- Second, to show the general applicability of the proposed MRs, we intend to utilize the proposed approach by testing a broad range of other clustering algorithms that are popular among both researchers and practitioners of the ML community

References

[1] Yang, S., Towey, D., & Zhou, Z. Q. (2019, May). Metamorphic exploration of an unsupervised clustering program. In *2019 IEEE/ACM 4th International Workshop on Metamorphic Testing (MET) (pp. 48-54)*. IEEE.

[2] Xie, X., Zhang, Z., Chen, T. Y., Liu, Y., Poon, P. L., & Xu, B. (2020). METTLE: a METamorphic testing approach to assessing and validating unsupervised machine LEarning systems. *IEEE Transactions on Reliability, 69(4), 1293-1322.*

[3] Basili, V. R. (1992). Software modeling and measurement: the Goal/Question/Metric paradigm.

Thank You ☺

Any Questions?