Evaluating Clustering Quality in Centroid-Based Clustering Using Counterfactual Distances

Georgios Vardakas(✉), Antonia Karra, Evaggelia Pitoura, and Aristidis Likas

Department of Computer Science and Engineering, University of Ioannina, 45110 Ioannina, Greece
{g.vardakas,a.karra,pitoura}@uoi.gr, arly@cs.uoi.gr

Abstract. Evaluating clustering quality is essential for selecting the optimal number of clusters and for comparing clustering algorithms. Several internal clustering quality indices have been proposed such as the Silhouette score and the Variance Ratio Criterion (VRC) with well-known advantages and limitations. In this work, we present the idea of counterfactuals to quantify cluster separation. Counterfactuals have been recently introduced in the context of clustering to quantify minimum modifications to be applied to a point of one cluster in order to be assigned to another cluster. We exploit counterfactual distances in the context of k-means clustering and define the separation between clusters considering such distances. Then we use the proposed separation measure to define a clustering quality score as the ratio of total separation over total intra-cluster variance of a clustering solution. We evaluated the effectiveness of the proposed score against Silhouette and VRC using various real-world digit datasets.

Keywords: Clustering quality · cluster separation · counterfactual distances · silhouette score

1 Introduction

Clustering is a fundamental task in unsupervised learning, where the goal is to group data points into meaningful clusters based on similarity. Evaluating clustering quality is essential for selecting the optimal number of clusters and for comparing clustering algorithms [5,7]. Among various clustering quality indices, the Silhouette score [9] and the Variance Ratio Criterion (VRC) [1] are the most widely used.

In this work, we focus on the idea of *counterfactuals* to quantify cluster separation. Counterfactuals have been widely adopted to explain classification decisions and have also been proposed recently for clustering problems [8,11]. Roughly speaking, considering a pair of clusters, a counterfactual of a data point in one cluster (source cluster) is its closest point in the other cluster (target cluster). Based on this definition, the counterfactual of point in the source cluster

S. Džeroski et al. (Eds.): DS 2025, LNAI 16090, pp. 222–236, 2025.
https://doi.org/10.1007/978-3-032-05461-6_15

is its closest point that lies in the *boundary* between the source and the target cluster. Thus, counterfactual distances can be used to quantify cluster separation.

This paper introduces a simple and efficient clustering quality criterion based on the ratio of total cluster separation to total intra-cluster variance. The proposed score, called *CFQ*, leverages the *counterfactual-based cluster separation measure* and scales linearly with dataset size, offering a computationally efficient alternative to existing criteria. We have tested and compared our criterion on the problem of selecting the number of clusters in the context of k-means clustering.

The structure of the paper is the following. Section 2 describes related work on clustering quality indices. Section 3 presents counterfactuals in the context of clustering illustrating that they can be used to measure the cluster separation in k-means clustering. Section 4 presents and explains the proposed counterfactual-based measure of cluster separation and defines the CFQ criterion for assessing clustering quality. Section 5 provides details and results of our experimental study with four real digits datasets. Finally, Sect. 6 provides conclusions and future research directions.

2 Internal Quality Indices

Cluster quality indices are used to evaluate the suitability of clustering solutions. A major distinction is between external and internal indices. *External indices* of clustering quality measure the similarity of a clustering solution to a reference solution, usually called ground truth solution. Typical measures of this type are clustering accuracy and normalized mutual information (NMI) [3]. It should be stressed that this similarity comparison involves only the labels of the two compared clustering solutions and does not involve information related to data points or the distance between them.

Nevertheless, clustering is an unsupervised learning problem; therefore, ground truth information is typically not provided to the clustering algorithms. Therefore, the usefulness of external quality indices is rather limited in practice. However, there is an imperative need for assessing the quality of clustering solutions based on the characteristics of the generated clusters in order to be able to select among possible alternative solutions. Such cluster solution evaluation is provided by *internal indices* which take into account only the labels of the clustering solution and information related to the data points that are clustered.

A critical use of internal indices is for the selection of the appropriate number of clusters in cases where this number is not provided by the user. The typical approach to tackle this important problem is to apply a clustering algorithm (e.g. k-means) on the same dataset several times, each time with a different value of number of clusters (k), evaluate the solution produced for each k using an internal quality score and select the solution with highest score.

Internal quality indices usually rely on the combination of two quantities: the first one is *intra-cluster variance* (high compactness) and the second is *separation*. A good clustering solution is typically characterized by high compactness (low intra-cluster variance) and high separation (high inter-cluster variance).

Therefore usually the ratio or the (normalized) difference between inter-cluster and intra-cluster variance is defined as the final quality score. Both compactness and separation can be measured and combined: i) either at point level, where a quality score for each data point is computed and the total score of the solution is the average of individual scores, or ii) at cluster level, where compactness and separation of each cluster are computed and then aggregated for all clusters. Moreover, the computations for compactness and separation could involve either pairwise distances between points or distances between points and cluster centroids or distances between cluster centers. In this work, we propose the use of more informative distance measures between points and clusters that are based on counterfactual distances, i.e. distances from cluster boundaries. In our experimental study, we compared against silhouette and VRC defined as follows.

2.1 Silhouette Coefficient

Assume a dataset X with n points partitioned into k clusters $C = \{C_1, \ldots, C_k\}$. Let $d(x_i, x_j)$ denote the distance between data points x_i and x_j. Silhouette [9] operates at the data point level and computes the distance of a point x to a cluster C_j as the average of distances of x to the points of C_j. A point $x_i \in C_I$ is considered as 'well-clustered' if its distance $a(x_i)$ from its own cluster is small and its minimum distance $b(x_i)$ to any other cluster is large:

$$a(x_i) = \frac{1}{|C_I| - 1} \sum_{x_j \in C_I, i \neq j} d(x_i, x_j) \tag{1}$$

$$b(x_i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{x_j \in C_J} d(x_i, x_j) \tag{2}$$

where $|C_I|$ and $|C_J|$ are cluster sizes. The silhouette score for x_i is given by:

$$s(x_i) = \frac{b(x_i) - a(x_i)}{\max\{a(x_i), b(x_i)\}}, \quad -1 \leq s(x_i) \leq 1 \tag{3}$$

The overall silhouette score $S(C)$ of clustering solution C is the average over all data points:

$$S(C) = \frac{1}{n} \sum_{i=1}^{n} s(x_i), \quad -1 \leq S(C) \leq 1 \tag{4}$$

Higher $S(C)$ values indicate good cluster assignments, while lower or negative values indicate a bad quality clustering solution.

2.2 Variance Ratio Criterion

The Variance Ratio Criterion [1] (also known as the Calinski-Harabasz Index) evaluates clustering quality by comparing the total between-cluster dispersion (nomimator) to the total within-cluster dispersion (denominator):

$$VRC(C) = \frac{\sum_{j=1}^{k} |C_j| \|\mu_j - \mu\|^2}{\sum_{j=1}^{k} \sum_{x \in C_j} \|x - \mu_j\|^2} \frac{n - k}{k - 1} \tag{5}$$

where $|C_k|$ is the size of cluster C_k, μ_k is the center of C_k, and μ *is the overall mean* of the dataset. Higher VRC values indicate better clustering, as they reflect greater between-cluster variability relative to within-cluster dispersion. A key limitation of VRC is that it measures cluster separation (nominator) using the variance of the cluster centers, which is a simplistic approach that may be ineffective for capturing complex cluster structures.

3 Counterfactual Distances

3.1 Counterfactuals for Classification

Counterfactual explanations are local explanations that have been widely used for classification problems. In essence, a counterfactual explanation provides suggestions on how the feature values of an example (called *factual*) should change in order for the modified example (called *counterfactual*) to be classified into a different class [4,12]. More specifically, let f be a classification model and $d(x, y)$ a distance function. Given an example (factual) y, its counterfactual explanation z is the data point closest to y whose outcome $f(z)$ differs from the prediction $f(y)$. More formally:

$$z = \arg\min_x d(x, y) \quad \text{s.t.} \quad f(z) \neq f(y). \tag{6}$$

Various search methods have been studied to solve the above problem ranging from gradient-based to genetic algorithms. Those methods are available in popular software libraries for generating counterfactuals in classification tasks supporting various distance metrics (such as L_1 and L_2 norms) and classification models f.

3.2 Counterfactuals for Clustering

We assume a clustering solution $C_1, ..., C_k$ with k clusters on a given dataset X. Given a factual y in cluster C_i and a distance function $d(x, y)$ its counterfactual explanation z is defined as the solution to the following constrained optimization problem [11]:

$$z = \arg\min_x d(x, y) \quad \text{s.t.} \quad C_\ell \neq C_i \tag{7}$$

In the above formulation, C_ℓ is the cluster label of z, taking into account the cluster assignment rule (e.g. the distance from cluster centers in the k-means case). Assuming a source and a target cluster, *the counterfactual of a given point (factual) is its closest point lying on the cluster boundary.* Therefore, the counterfactual distance of a point to the target cluster, provides an effective and optimal way to *quantify distance of a point from a cluster.* By summing (or averaging) the counterfactual distances of points, we can compute a measure of separation between two clusters. This idea is exploited in our proposed clustering quality score.

In [11] we have proposed ways for computing counterfactuals for k-means and Gaussian clustering specifically targeting the L_2 distance norm between factual and counterfactual point. The counterfactual computation for k-means clustering is simple and analytical as explained next.

3.3 Counterfactuals for k-Means Clustering

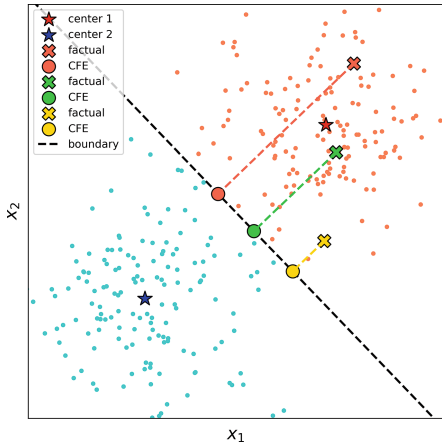

Fig. 1. Counterfactual computation for k-means clustering.

Assuming two clusters C_j and C_l with centers μ_j and μ_l respectively, the cluster boundary set B contains the points z for which:

$$|z - \mu_j|^2 = |z - \mu_l|^2 \tag{8}$$

This equation defines a *hyperplane* that is perpendicular at the middle (point $m_{lj} = (\mu_j + \mu_l)/2$) to the vector $n_{jl} = \mu_l - \mu_j$ connecting the two centers. We refer to the cluster boundary as *separating hyperplane*.

Assuming a point x (factual) of cluster C_j and the L_2 distance measure, its closest counterfactual point $cf_{jl}(x)$ with respect to cluster C_l is computed analytically and corresponds the projection of x to the separating hyperplane:

$$cf_{jl}(x) = x - \frac{(x - m_{lj})^\top n_{lj}}{|n_{lj}|^2} n_{lj} \tag{9}$$

Moreover, the Euclidean distance between x and its counterfactual $cf_{jl}(x)$ is given by the *projection distance*:

$$d_{jl}(x) = \frac{|(x - m_{jl})^\top (\mu_l - \mu_j)|}{\|\mu_l - \mu_j\|}, \tag{10}$$

Therefore, in the k-means case, the counterfactual is computed *analytically* using the two clusters centers μ_j, μ_l and the factual x. In Fig. 1 we provide illustrative examples of counterfactuals generated considering a synthetic 2-d dataset clustered using k-means. It can be observed, the distances between cluster points and their counterfactuals (which lie on the cluster boundary), can be used to define effective measures of *cluster separation*.

4 Counterfactual-Based Quality Score

Assuming a dataset $X = \{x_1, \ldots, x_n\}$, we propose a new criterion for evaluating the quality of a clustering solution $C = \{C_1, \ldots, C_k\}$ with k clusters:

$$CFQ(C) = \frac{\sum_{j=1}^{k} \text{separation}(j)}{\sum_{j=1}^{k} \text{variance}(j)}, \tag{11}$$

where variance(j) is the total squared distance of points in cluster j to their centroid μ_j (intra-cluster variance), and separation(j) is the minimum distance between cluster j and any other cluster, computed using counterfactual distances as follows.

Given the centroids μ_j and μ_l of clusters C_j and C_l, we can compute the Euclidean counterfactual distance $d_{jl}(x)$ of a point $x \in C_j$ from cluster C_l using Eq. 9. We define the distance $Sep(j, l)$ of cluster C_j to C_l as the sum of squared counterfactual distances of the points of C_j to cluster C_l:

$$Sep(j, l) = \sum_{x_i \in C_j} d_{jl}^2(x_i) \tag{12}$$

Note that $Sep(j, l)$ is not symmetric, i.e. $Sep(j, l) \neq Sep(l, j)$. Then, by transferring the silhouette idea to the cluster level, we define the separation $S(j)$ of cluster C_j as its minimum separation to any other cluster:

$$S(j) = min_{l, l \neq j} Sep(j, l) \tag{13}$$

The total separation $TS(C)$ of the clustering solution is computed as the sum of individual cluster separations:

$$TS(C) = \sum_{j=1}^{k} S(j) \tag{14}$$

It should be noted that the above sum includes one counterfactual distance term for each data point. In order to measure the total variance $TV(C)$ of the clustering solution, we use the typical clustering error minimized by the k-means algorithm:

$$TV(C) = \sum_{i=1}^{n} \sum_{j=1}^{k} I(x_i \in C_j) ||x_i - \mu_j||^2 \tag{15}$$

Note that, as in the case with separation measure, one distance term for each point is also included in the total variance measure. Therefore both TS(C) and TV(C) are of the same scale and we define our clustering quality measure (CFQ) as the ratio of the two terms:

$$CFQ(C) = \frac{TS(C)}{TV(C)} \tag{16}$$

Based on the above definition, higher CFQ values indicate better clustering quality. A notable advantage of CFQ is its *linear computational complexity*: $O(nkd)$ where d is the dimension of the data points. In contrast, the computational complexity of silhouette is $O(n^2)$, thus for large datasets, silhouette is usually approximately computed using randomly selected data subsets.

Figure 2 presents an example where a 2-d synthetic dataset with 5 clusters has been partitioned using k-means with k from 2 to 5. The silhouette, VRC and CFQ scores of each solution are also presented. It can be observed that CFQ selects the correct solution ($k = 4$) evaluating this solution with the highest CFQ score. In contrast, silhouette selects $k = 2$, while VRC selects $k = 5$.

The silhouette score has a tendency to favor simpler partitions with fewer clusters, showing a bias toward *underclustering*. In particular, when well separated groups contain finer internal structures, silhouette may fail to distinguish the subclusters within each group. Instead, it often favors merging them into a single cluster, overlooking meaningful distinctions in locally separated regions.

The proposed counterfactual-based criterion (CFQ) evaluates clustering quality by comparing cluster separability to intra-cluster compactness. While it captures meaningful separations more robustly than traditional metrics, it exhibits a tendency to *overclustering*. In CFQ, the score improves when clusters become more compact (due to splitting) while maintaining sufficient separation. As a result, CFQ tends to favor solutions where large, dense clusters are artificially divided into multiple subclusters, even when such splits do not correspond to meaningful structural differences. This occurs because the decrease of the total variance term is greater than the decrease of the separation term, leading to an inflated CFQ score.

Figure 3 presents another example using a 2-d synthetic dataset with 4 clusters which has been partitioned using k-means with k from 2 to 8. Only the solutions for $k = 2$ and $k = 7$ are presented. CFQ overclusters the dataset selecting the solution with $k = 7$ clusters, while silhouette underclusters the dataset selecting $k = 2$. VRC also selects $k = 7$. Silhouette and CFQ tendencies for underclustering and overclustering, respectively, highlight the need for caution when relying solely on a single internal validation metric to guide clustering decisions.

In what concerns VRC, in practice, it often proves less sensitive to local cluster structures compared to CFQ and Silhouette. Its restrictive reliance on global variance and distances between centers limits its effectiveness when clusters differ significantly in shape, size or density.

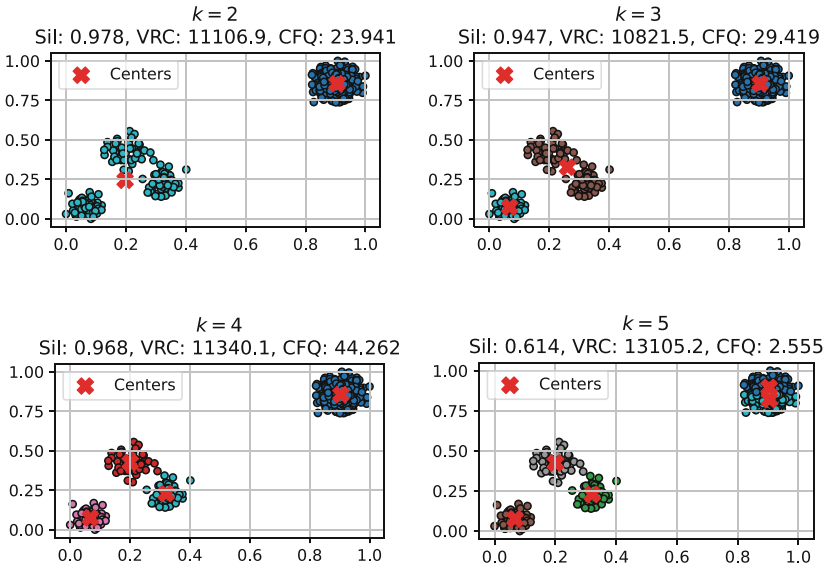

Fig. 2. k-means clustering results for different values of k using a synthetic 2-d dataset. CFQ attains maximum value at the correct solution $k = 4$. Silhouette attains maximum at $k = 2$ (underclustering). VRC attains maximum at $k = 5$ (overclustering).

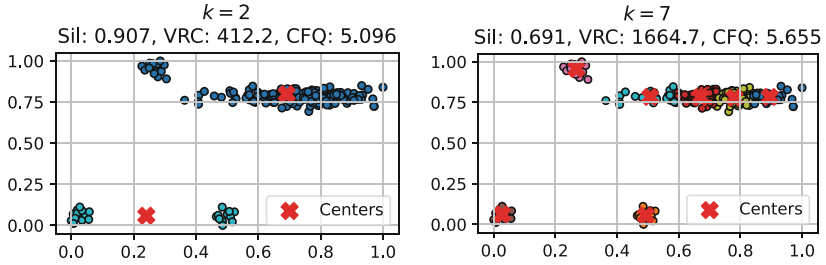

Fig. 3. k-means clustering results for $k = 2$ and $k = 7$ using a synthetic 2-d dataset. Silhouette suggests $k = 2$ (underclustering). CFQ and VRC suggest $k = 7$ (overclustering).

5 Experimental Results

The aim of our empirical study is to comparatively assess the effectiveness of the proposed CFQ criterion in number of clusters estimation: selecting the best among several clustering solutions provided by applying the k-means algorithm on the same dataset for various values of the number of clusters k. We compare CFQ to both silhouette (with squared Euclidean pairwise distances) and VRC.

5.1 Datasets

We considered four well-known 10-class real datasets related to numerical digits (0–9). Those datasets have been selected since they are known to demonstrate substantial clustering structure, i.e. the examples of each digit class tend to form relatively compact clusters that are discernible to some degree. Table 1 summarizes the four digit datasets that we used in our experimental evaluation.

The OptDigits dataset consists of grayscale 8×8 images of handwritten digits. Each 16-dimensional example in the Pendigits dataset contains the (x, y) coordinates of eight pen points on a digital screen. USPS contains 16×16 images of handwritten images. The MNIST embeddings dataset contains 10-dimensional representations of grayscale 64×64 images of handwritten digits, obtained from a trained autoencoder [10]. In all datasets we used min-max normalization as a preprocessing step to map the features of each data point to the $[0, 1]$ interval.

Table 1. The real world digits datasets used in the experiments. n is the number of data instances, d is the dimensionality.

Dataset	n	d	Source
OptDigits	1797	64	[2]
Pendigits	10992	16	[2]
USPS	9298	256	[6]
MNIST (embeddings)	60000	10	[2]

5.2 Experimental Protocol and Results

For each of the four datasets a series of 100 experimental trials we conducted. At each trial, a data subset is first created as follows:

- We randomly select the number of classes k_{actual} as an integer between 2 and 10. Note that this is the ground truth number of clusters of the trial.
- We randomly select k_{actual} different digit classes (from 0–9).
- For each selected digit, we randomly specify the number of examples per class between a minimum value equal to 20 and a maximum value equal to 1000. If the number of available examples of a class is less than 1000, then we set our maximum value equal to this number.
- For each selected class, we randomly select (without replacement) a subset of examples of size equal to the previously specified numbers.

In the above way, a data subset is generated in each trial containing a random number of randomly selected digit classes and with the number of examples per digit varying from possibly very low (near 20) to possibly very high (near 1000). Therefore, some subsets are expected to be highly imbalanced. For each

generated subset, the actual number of clusters is known as well as the ground truth labeling of the examples. This information is used to evaluate the clustering solutions selected by the three compared quality scores.

Table 2. NMI statistics over 100 trials for each dataset.

Dataset	Method	Mean	Std
Optdigits	CFQ	0.711	0.118
	SIL	0.725	0.121
	VRC	0.477	0.204
Pendigits	CFQ	0.641	0.135
	SIL	0.629	0.149
	VRC	0.543	0.178
MNIST Embeddings	CFQ	0.712	0.138
	SIL	0.691	0.174
	VRC	0.542	0.194
USPS	CFQ	0.561	0.149
	SIL	0.465	0.207
	VRC	0.394	0.179

After subset generation, the following experiment was conducted at each trial: we applied the k-means algorithm to this subset for values of k from 2 to 15. The solution for each k was evaluated using the three criteria (CFQ, Silhouette, VRC) and for each criterion the solution with highest score was selected as the final solution. The solution selected by each criterion was evaluated i) based on the accuracy of k estimation (absolute difference from actual k) and ii) the NMI [3] of the solution with respect to the ground truth solution.

For each digit dataset, we repeated the above protocol for each of the 100 trials and present two plots. The first plot provides the NMI score for each trial of the k-means solution selected by CFQ (blue line), silhouette (SIL) (orange line) and VRC (green line). The NMI of the k-means solution with ground truth k (k_{actual}) is also presented (black line). The second plot for each dataset presents for each trial the difference $(k^* - k_{actual})$ between the value $k^\star$ of the solution selected by each score and the ground truth k_{actual}. The line colors are the same as in the NMI plot. Obviously, the ground truth solution (black line) has a constant difference equal to zero. Therefore solutions above the black horizontal line correspond to overclustering and below it to underclustering. Figure 4 and 5 provide results for Optdigits, Fig. 6 and 7 for Pendigits, Fig. 8 and 9 for MNIST embeddings, while Fig. 10 and 11 provide results for USPS.

Table 2 provides for each dataset the average and standard deviation over the 100 trials of the NMI scores of the k-means solutions selected by each of the three scores. Table 3 presents for each dataset the number of trials for which CFQ solution was found superior, equal or inferior to the silhouette (SIL) or VRC solution in terms of NMI (higher is better) and in terms of estimation of the actual number of clusters $|k^\star - k_{actual}|$ (lower is better).

From the results in Table 3 (columns "Tie") it can be observed that in about half of the 400 trials with all datasets, CFQ and SIL provide solutions of equal quality both in terms of NMI and estimation of the true number of clusters. Therefore, they can be considered as equal strength competitors in general. In what concerns VRC, it is clearly inferior to the other scores in all four datasets.

Results for the Optdigits dataset indicate a moderate superiority of SIL compared to CFQ both in terms of NMI and k estimation. while the opposite holds for USPS dataset. In MNIST embeddings CFQ appears to be slightly superior, while in Pendigits SIL is slightly better in terms of k estimation and CFQ in terms of NMI. The inspection of the k estimation plots for all datasets reveals the tendency of CFQ to overstimate the actual number of clusters, while SIL and VRC tend to underestimate this number, with SIL providing estimations closer to ground truth.

Table 3. Number of trials for which CFQ was superior ("CFQ" column), inferior ("Other" column) or equal ("Tie" column) to SIL or VRC in terms of NMI and estimation of true number of clusters k_{actual}.

Dataset	Comparison	NMI			$\lvert k^* - k_{actual}\rvert$		
		CFQ	Other	Tie	CFQ	Other	Tie
Digits	CFQ vs SIL	20	34	46	9	41	50
	CFQ vs VRC	80	16	4	50	29	21
Pendigits	CFQ vs SIL	26	14	60	13	23	64
	CFQ vs VRC	61	12	27	47	18	35
MNIST Embeddings	CFQ vs SIL	32	23	45	27	21	52
	CFQ vs VRC	76	6	18	63	12	25
USPS	CFQ vs SIL	43	19	38	38	20	42
	CFQ vs VRC	59	5	36	49	9	42

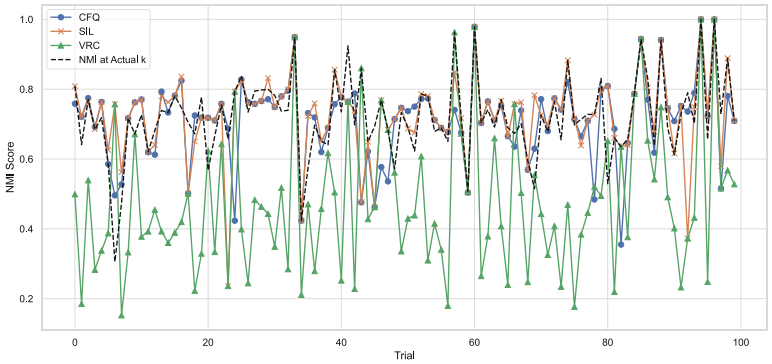

Fig. 4. Optdigits: NMI across trials for different quality scores.

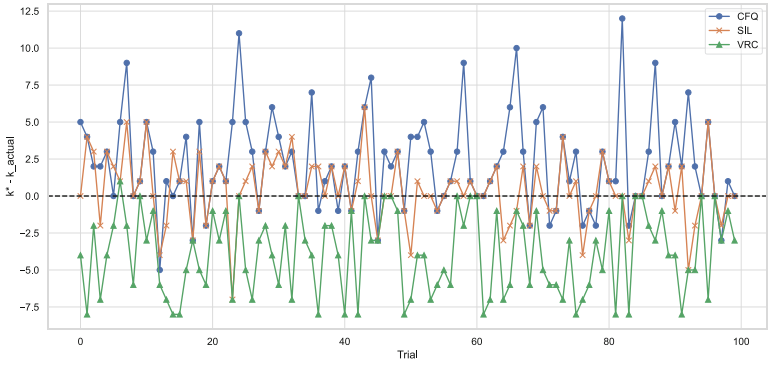

Fig. 5. Optdigits: Error between predicted and true number of clusters ($k^* - k_{actual}$) across trials for different quality scores.

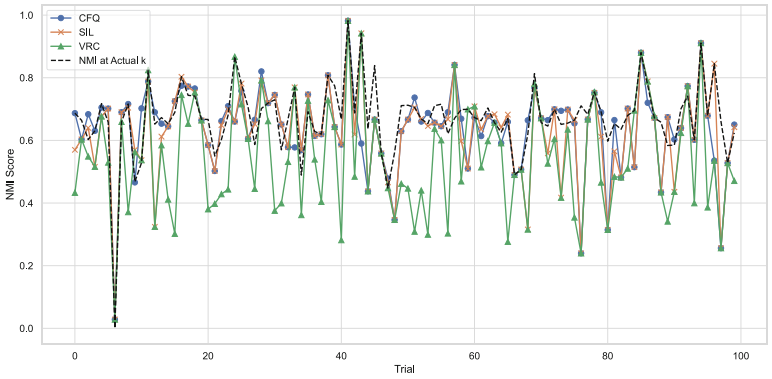

Fig. 6. Pendigits: NMI across trials for different quality scores.

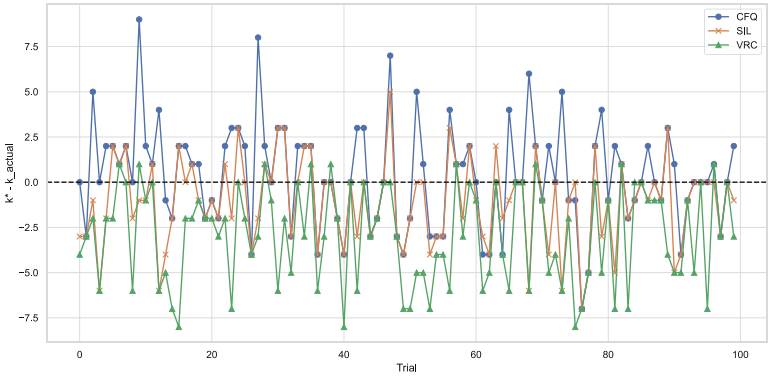

Fig. 7. Pendigits: Error between predicted and true number of clusters ($k^* - k_{actual}$) across trials for different quality scores.

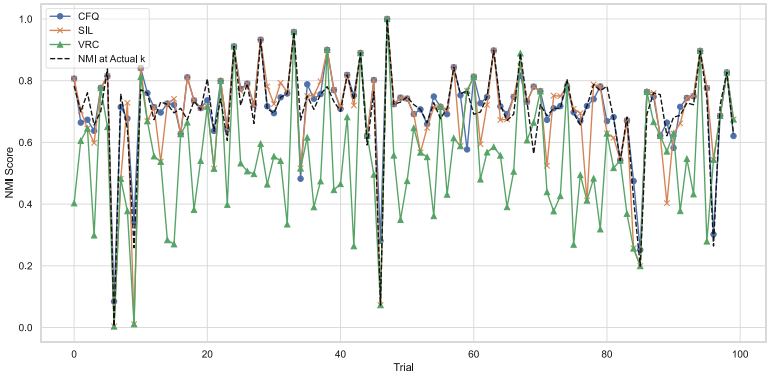

Fig. 8. MNIST (embeddings): NMI across trials for different quality scores.

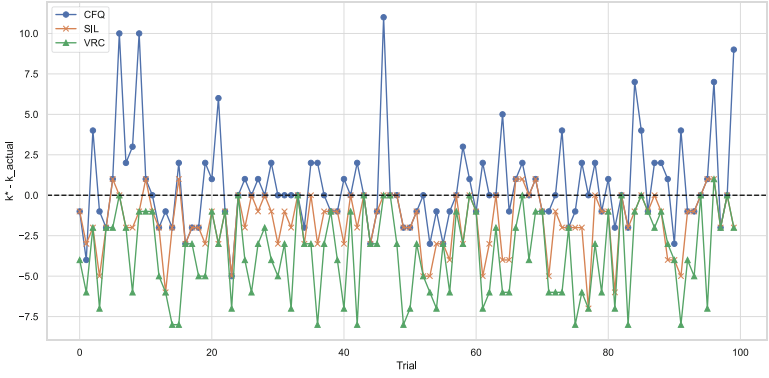

Fig. 9. MNIST (embeddings): Error between predicted and true number of clusters ($k^* - k_{actual}$) across trials for different quality scores.

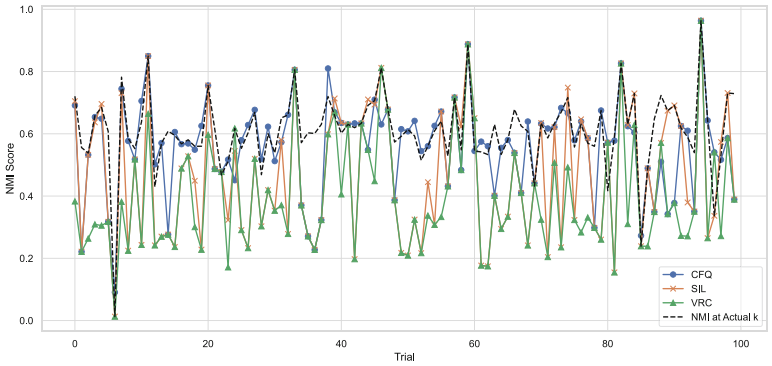

Fig. 10. USPS digits: NMI across trials for different quality scores.

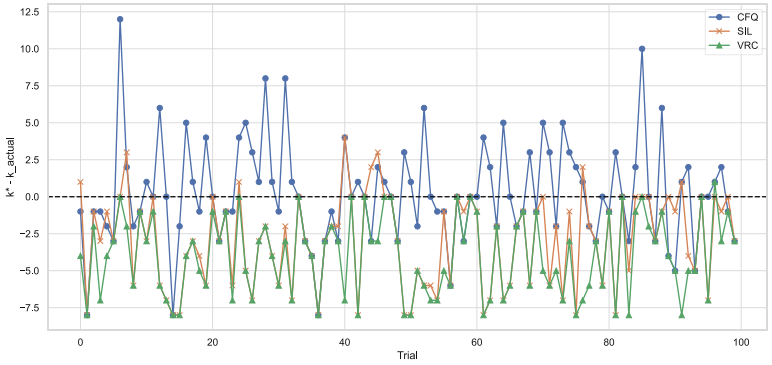

Fig. 11. USPS digits: Error between predicted and true number of clusters ($k^* - k_{actual}$) across trials for different quality scores.

6 Conclusions

We introduced a new approach to quantifying cluster separation by leveraging counterfactual distances. Counterfactuals have been recently used in clustering to represent the minimal changes required for a point from one cluster to be reassigned to another. Building on this idea, we used counterfactual distances within the framework of k-means clustering to define a separation measure between clusters. We also proposed a clustering quality score (CFQ), defined as the ratio of total separation to total intra-cluster variance. We assessed the effectiveness of CFQ in comparison to established metrics such as silhouette and VRC across several digit datasets.

Future work could focus on a more elaborate empirical evaluation of CFQ both in real-world clustering applications (e.g. face clustering, video summarization, text clustering) as well as in assessing its robustness in the presence of noise and outliers. Moreover, we aim to employ counterfactual distances to measure separation in Gaussian clustering. In such a case the cluster boundary is non-

linear and counterfactuals are computed by solving a non-linear equation with a single parameter [11].

Acknowledgment. The research project is implemented in the framework of H.F.R.I. call "Basic research Financing (Horizontal support of all Sciences)" under the National Recovery and Resilience Plan "Greece 2.0" funded by the European Union - NextGenerationEU (H.F.R.I. ProjectNumber: 15940).

Declarations. Conflict of interest The authors declare no conflict of interest.

References

1. Caliński, T., Harabasz, J.: A dendrite method for cluster analysis. Commun. Stat.-Theory Methods **3**(1), 1–27 (1974)
2. Dua, D., Graff, C.: UCI machine learning repository (2017). http://archive.ics.uci.edu/ml
3. Estévez, P.A., Tesmer, M., Perez, C.A., Zurada, J.M.: Normalized mutual information feature selection. IEEE Trans. Neural Netw. **20**(2), 189–201 (2009)
4. Guidotti, R.: Counterfactual explanations and how to find them: literature review and benchmarking. Data Min. Knowl. Disc. **38**(5), 2770–2824 (2024)
5. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. J. Intell. Inf. Syst. **17**(2–3), 107–145 (2001)
6. Hull, J.J.: A database for handwritten text recognition research. IEEE Trans. Pattern Anal. Mach. Intell. **16**(5), 550–554 (1994)
7. Ikotun, A.M., Habyarimana, F., Ezugwu, A.E.: Cluster validity indices for automatic clustering: a comprehensive review. Heliyon **11**(2), e41953 (2025)
8. Karra, A., Vardakas, G., Pitoura, E., Likas, A.: Generating counterfactual explanations for clustering models based on their equivalence to classification models. In: IFIP International Conference on Artificial Intelligence Applications and Innovations. Springer, Cham (2025)
9. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987)
10. Van Der Maaten, L.: Learning a parametric embedding by preserving local structure. In: Artificial Intelligence and Statistics, pp. 384–391. PMLR (2009)
11. Vardakas, G., Karra, A., Pitoura, E., Likas, A.: Counterfactual explanations for k-means and gaussian clustering. arXiv preprint arXiv:2501.10234 (2025)
12. Verma, S., Boonsanong, V., Hoang, M., Hines, K., Dickerson, J., Shah, C.: Counterfactual explanations and algorithmic recourses for machine learning: a review. ACM Comput. Surv. **56**(12), 1–42 (2024)