Linkage Criteria for Agglomerative Clustering based on Counterfactual Distances

Georgios Vardakas
Dept. of Computer Science & Engineering
University of Ioannina
Ioannina, Greece
g.vardakas@uoi.gr

Antonia Karra
Dept. of Computer Science & Engineering
University of Ioannina
Ioannina, Greece
a.karra@uoi.gr

Evaggelia Pitoura
Dept. of Computer Science & Engineering
University of Ioannina
Ioannina, Greece
pitoura@uoi.gr

Aristidis Likas
Dept. of Computer Science & Engineering
University of Ioannina
Ioannina, Greece
arly@cs.uoi.gr

***Abstract*—Agglomerative clustering is a widely used hierarchical clustering method that relies on linkage criteria (quantifying cluster separation) to iteratively merge the closest clusters each time. Traditional linkage methods, such as single link and average link are quite simple and in several cases may not adequately capture the true separation between clusters, especially in the case of noise or varying cluster densities. In this work, we propose new measures of cluster separation grounded in counterfactual distances. Assuming a pair of clusters, the counterfactual of a point in one cluster is its closest point in the other cluster. We show that counterfactuals effectively characterize the cluster borders. More specifically, the counterfactuals of points from one cluster effectively characterize the frontier of the other cluster. Consequently, distances between mutual counterfactual points provide a more reliable and representative estimate of inter-cluster separation. Based on this insight, we introduce the Mutual Counterfactual (MCF) and Iterative Mutual Counterfactual (IMCF) linkage criteria, which consider only distances between mutual counterfactuals. We integrate the proposed linkages into the standard agglomerative clustering algorithm and perform comparative experiments against conventional linkage methods with promising performance results.**

***Index Terms*—agglomerative clustering, cluster separation, linkage criteria, counterfactuals.**

I. INTRODUCTION

Hierarchical clustering is a key method in unsupervised learning and data mining, with agglomerative hierarchical clustering being one of the most intuitive and commonly used approaches [1]. This method starts by considering each data point as its own cluster and progressively merges the most similar clusters step by step. The process continues until all data points are combined into a single overarching cluster, forming a nested structure known as a dendrogram.

The dendrogram offers a visual representation of the data's structure at multiple levels of granularity. By selecting a specific cut-off point on the dendrogram, one can extract clusters at varying resolutions, making it especially useful for exploring complex datasets. Unlike partitional clustering techniques, which require the number of clusters to be specified beforehand.

However, despite its straightforward nature, agglomerative clustering does come with certain limitations, such as significant computational demands for large datasets and vulnerability to noise and outliers. Even so, it remains a powerful method, particularly when interpretability and hierarchical insights are essential.

In order for typical agglomerative algorithms to be applied, the so called *linkage criterion* should be defined that specifies how the distances between clusters are computed. The most well-known linkage strategies are [2], [3]:

- *Single Linkage* (SL): Computes the minimum distance between any two points in different clusters. It can capture elongated shapes but suffers from the chaining effect, which can lead to poorly separated clusters. It is also very sensitive to the existance of noise points between the clusters.
- *Average Linkage* (AVL): Calculates the average of all pairwise distances between points in the two clusters. It provides a balance between single and complete linkage and often gives reasonable performance across a variety of data types. However, it demonstrates a tendency to split large compact clusters.
- *Complete Linkage* (CL): Uses the maximum distance between points in different clusters. It produces compact clusters and is less sensitive to noise but can fragment large, spread-out clusters.

The above linkage criteria are based on the pairwise distances between data points, therefore they require only the distance matrix of the dataset as input, and not the data points. In addition they allow for any type of pairwise distance functions to be used. Additional linkage criteria have been proposed that require the data points as inputs and are restricted to Euclidean pairwise distances:

- *Centroid Linkage*: Merges clusters based on the Euclidean distance between their centroids.
- *Ward's Method*: Seeks to minimize the total within-cluster variance. It merges clusters that result in the smallest increase in total intra-cluster variance and tends to form clusters of similar size and shape.

979-8-3315-5636-5/25/$31.00

As mentioned above, those widely adopted linkage criteria attempt to quantify the separation between clusters and demonstrate advantages and drawbacks depending on the characteristics of the clusters (compact or elongated), the existence of noisy points between the clusters etc. A natural observation is that cluster separation is directly related to the distance of the points that lie on the *cluster borders*. From the above typical linkages, only the single link criterion focuses on the cluster borders, but this is done in a rather naive way since it considers only a pair of borderline points from the two clusters. This is by no means sufficient to accurately quantify the distance between borderline points in the two clusters.

In this work, we introduce the idea of *counterfactuals* to determine borderline cluster points and quantify cluster separation. Counterfactuals have been widely adopted to explain classification decisions and have also been proposed recently for clustering problems [4], [5]. 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). Such counterfactual point can belong to the given set of target cluster points or can be a synthetic one. We focus on counterfactuals of the first type in this work and demonstrate that they can be used to represent the *cluster frontiers*, ie. set of points on the cluster borders. An illustrative example is presented in Fig. 1a. The annotated blue points constitute counterfactuals of the points in the orange cluster, while the annotated orange points constitute counterfactuals of the points of the blue clusters. It can be observed that the counterfactuals (annotated points) sufficiently represent the frontier of each cluster.

Next, we define linkage criteria based on distances from counterfactuals (called *counterfactual linkages*) and employ these linkages as cluster separation measures for agglomerative clustering. The proposed linkages can be considered as extensions to single link and average link. They do not require the data vectors as input and work directly on the values of the pairwise distances among the data points. They been evaluated in terms of clustering performance on several real and synthetic datasets and compared against the typical linkage criteria.

The structure of the paper is the following. In Section II we describe the notion of counterfactuals focusing on their use in the clustering context and illustrating that they can be used to represent the cluster borders. Section III presents and explains the proposed counterfactual linkages for quantifying the separation between clusters. Section IV focuses on agglomerative clustering using the proposed linkage criteria, while Section V provides details and results of the comparative experimental study. Finally, Section VI provides conclusions and several future research directions.

II. COUNTERFACTUAL EXPLANATIONS

A. 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 feauture values of an example (called factual) should change in order for the modified example (called counterfactual) to be classified into the desired (target) class [6]–[8]. 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 a data point close to y whose outcome $f(z)$ differs from the prediction $f(y)$. More formally:

$$z = \arg\min_{x} d(x, y) \quad \text{s.t.} \quad f(z) \neq f(y). \tag{1}$$

Assuming a target class c^*, the typical approach to tackle the above constraint minimization problem is to minimize the following objective function [6]:

$$z = \arg\min_{x} \; d(x, y) + \lambda \cdot \mathcal{L}(f(x), c^*), \tag{2}$$

where:

- $\mathcal{L}(f(x), c^*)$ is a loss function penalizing the difference between $f(x)$ and c^*
- λ is a regularization parameter that balances between minimal changes and achieving the desired outcome.

Various search methods have been studied to solve the above problem ranging from gradient-based [9], [10], [11], [12] to genetic algorithms [13], [14]. Those methods are available in popular software libraries for generating counterfactuals in classification tasks. They offer support for various distance metrics (such as L_1 and L_2 norms), classification models f, and loss functions $\mathcal{L}$. Additionally, they provide options to define feasibility constraints, including the specification of actionable and immutable features, as well as restrictions on the allowable ranges of feature values.

B. Counterfactuals for Clustering

We assume a clustering solution with $C_1, ..., C_M$ with M clusters on a given dataset. In order to compute a counterfactual explanation z for a given example (factual) y, we need to specify a distance function $d(x, y)$. Given a factual y of cluster C_k, its counterfactual explanation z is defined as the solution to the following constrained optimization problem:

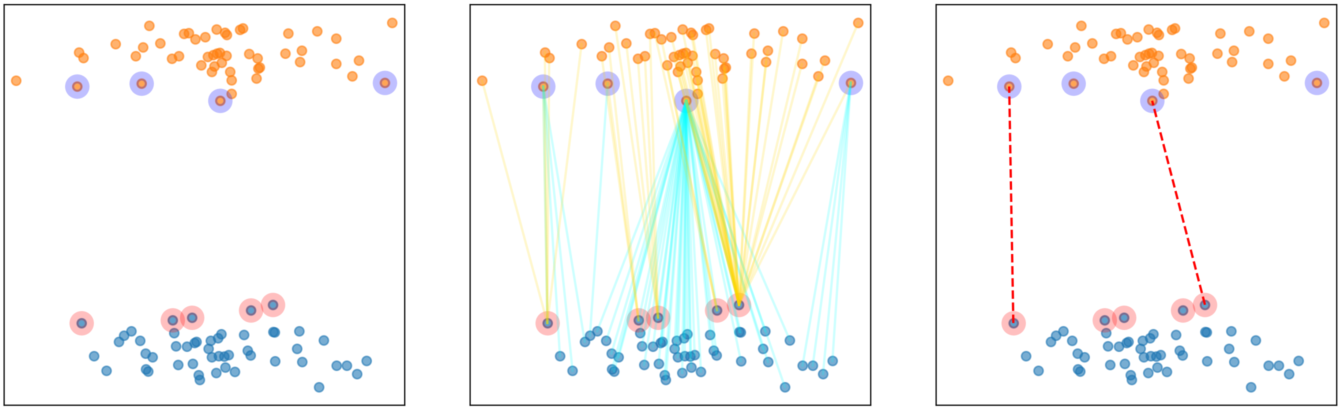
$$z = \arg\min_{x} d(x, y) \tag{3}$$

subject to the constraint:

$$C_\ell \neq C_k \tag{4}$$

In the above formulation, C_ℓ is the cluster label of z, taking into account the cluster assignment rule (eg. the distance from cluster centers in the k-means case). It should be noted in the general formulation above the counterfactual z is not restricted to be a point of the dataset.

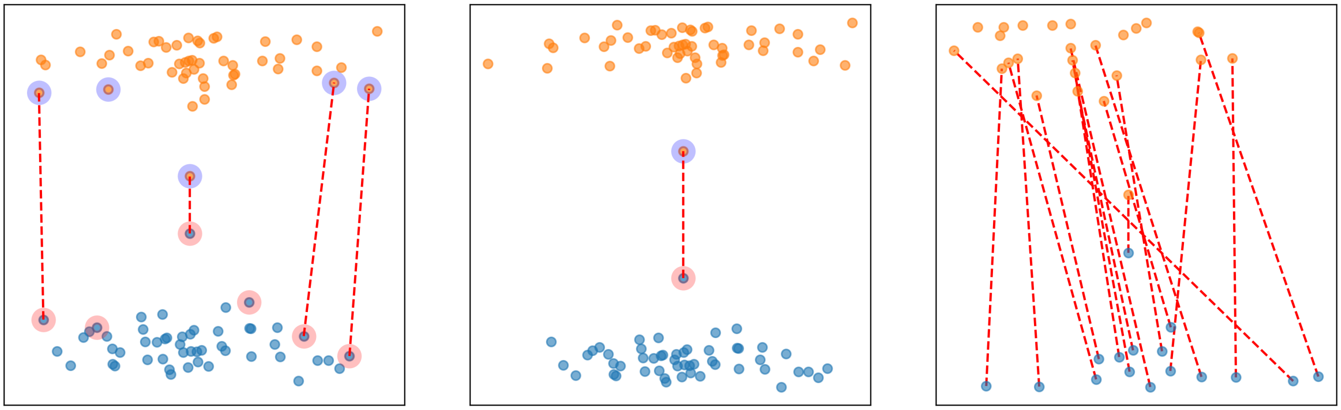
In [4] we have introduced a method for computing optimal counterfactuals for k-means and Gaussian clustering specifically targeting the L_2 distance norm between factual and counterfactual instances. In [5] we have proposed a methodology that builds on previous work on counterfactuals for classification and is applicable to any distance measure. In our experiments, we demonstrate its effectiveness by presenting results using the L_2 distance.



(a) Counterfactual points (marked circles) are located on the cluster borders.

(b) Counterfactual links connecting each point with its corresponding counterfactual point.

(c) Mutual counterfactual links.



(d) Mutual counterfactual links for a synthetic dataset with noise.

(e) Mutual counterfactual links for another synthetic dataset with noise, but with higher cluster separation.

(f) Iterative mutual counterfactual links.

Fig. 1: Illustration of the proposed counterfactual based linkages.

As mentioned previously, in the general problem formulation the domain of x is the whole data space, thus counterfactuals typically are not points of the dataset. In current work we will restrict our counterfactual search to the set of data points. Since we will focus on pairs of clusters (let C_k and C_ℓ) the counterfactual CF_i of given data point $x_i \in C_k$ is defined as:

$$CF_i = \arg\min_{z \in C_\ell} d(x_i, z) \tag{5}$$

Essentially, *the counterfactual of a data point is its nearest neighbor belonging to the other cluster*.

For a given cluster pair, a nice property of the set of counterfactuals is that they lie on the cluster borders; thus, they define the *cluster frontier*. More specifically, the counterfactuals of the points of one cluster define the frontier of the other cluster. Fig. 1a, already explained in the Introduction section, provides an illustrative example of this property.

Thus, the distances $d(x_i, CF_i)$ between cluster points and their counterfactuals (which belong to the other cluster), can be used to define effective measures of *separation* between the two clusters. Such linkage criteria are presented in the next section.

III. COUNTERFACTUAL LINKAGES FOR CLUSTER SEPARATION

Assuming that a given pair of clusters C_k and C_ℓ and the pairwise distances $d(x_i, x_j)$ between the points of the two clusters, our aim is to define a cluster separation measure based on counterfactuals. By computing for each data point x_i, its nearest neighbor in the other cluster (CF_i), we define the (x_i, CF_i) pairs. We will call such a pair as *counterfactual link* (CF link). The average distance of CF links (called $CFlinkage$) expresses the average distance of all data points from their corresponding counterfactuals and constitutes a natural measure of the separation between the two clusters (ie. a linkage criterion). In Fig. 1b the CF links are demonstrated and the CF linkage is the average distance of those links.

It should be noted that the CF linkage can be considered as a *special case of average linkage*. The average linkage takes into account the distances of a point to all points in the other cluster, while the proposed linkage is more selective, focusing on points on the cluster frontiers; thus, it is expected to be more informative in terms of cluster separation. In general CF linkage is expected to demonstrate an agglomerative clustering behavior analogous to average linkage, however providing improved performance since it is restricted to counterfactual links only.

Moving further, if for a pair (x_i, x_j) of points belonging to different clusters, it holds that each one constitutes the counterfactual of the other, ie.

$$x_i = CF_j \text{ and } x_j = CF_i \tag{6}$$

then we will say that the pair (x_i, x_j) constitutes a *mutual counterfactual link* (MCF link). The average distance between MCF links defines another linkage criterion called *MCF linkage*. MCF is more selective compared to CF linkage, since it considers only distances between counterfactual points. Therefore, it focuses only on distances between points that belong to the cluster frontiers. The MCF linkage can be considered as an *extension to singe linkage*, since the single link (pair of closest points between the clusters) is also an MCF link. However, MCF linkage typically takes into account more pairs of points on the cluster borders. Illustative examples are presented in Fig. 1c and Fig. 1d.

In Fig. 1d, a pair of noise points exist between the cluster borders. In this case, the single link criterion will erroneously consider cluster separation as the short distance between those two points. On the other hand, the MCF linkage will take into account all four MCF links, providing a quite accurate measure of the cluster separation.

An example where the MCF link reduces to a single link is presented in Fig. 1e. In this case, the two clusters are quite far apart (compared to Fig. 1d), thus the noise point of one cluster constitutes the counterfactual of all points of the other cluster. Thus, the set of MCF links contains only the pair of noise points and fails to adequately quantify cluster separation.

An approach to tackle the above problem, reduce dependence on noise points, and allow more MCF links to contribute to computing cluster separation, we introduce another counterfactual linkage called *Iterative Mutual Counterfactual* (IMCF) linkage. The IMCF link set is built iteratively by finding the MCF links at each iteration, adding them to the link set, removing the points corresponding to the selected MCF links and proceeding to the next iteration until no MCF links exist. More specifically, the algorithmic steps for computing the IMCF linkage criterion for cluster separation are presented in Algorithm 1.

Algorithm 1 Iterative Mutual Counterfactual Linkage (IMCF)

```
Require: $D = [d(k,l)]$ (pairwise point distances)
Require: $c_i$ (cluster $i$)
Require: $c_j$ (cluster $j$)
IMCF_links ← ∅
Compute MCF links between $c_i, c_j$
while MCF links are available do
    IMCF_links ← IMCF_links ∪ MCF links
    Update $c_i$ and $c_j$ by eliminating the points of MCF links
    Recompute MCF links of the updated clusters
end while
AvgIMCFDist = $\frac{1}{|\text{IMCF\_links}|} \sum_{(k,l) \in \text{IMCF\_links}} d_{k,l}$
return AvgIMCFDist
```

The above procedure returns a number of counterfactual links that is typically higher than the MCF linkage but lower than the CF linkage. In this way, it ensures robustness to the existence of noise points between the clusters, while still putting more emphasis on points in the cluster frontiers. Fig. 1f illustrates the links that contribute to the computation of IMCF linkage criterion. It can be observed that by removing the two noise points in the first iteration of the method, more CF links contribute to linkage computation. The IMCF linkage can be considered as *special case of average linkage* and *CF linkage*, with an appropriately selected set of links contributing to measure cluster separation.

IV. AGGLOMERATIVE CLUSTERING USING COUNTERFACTUAL LINKAGES

The previously introduced counterfactual linkage criteria can be implemented into the standard agglomerative approach: we start from singleton clusters and at each step we merge the closest pair of clusters (based on the employed linkage criterion) until the desirable number of clusters is obtained.

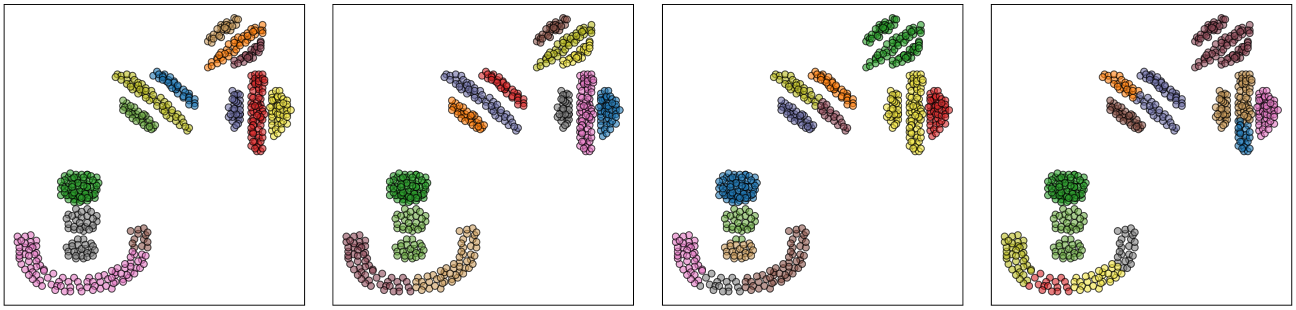
A subtle issue to be noted is that in standard linkages the computation of the separation between a pair of clusters does not make use of any information regarding distances to points in the remaining clusters. On the contrary, the computation of counterfactual linkages makes use of the distance values of a point to all other points in order to select the counterfactual for that point, ie. its nearest point that belongs to a different cluster. As a result, some points of a given cluster may have counterfactual links to points in one cluster, while some other points of the given cluster may have counterfactual links to points in a different cluster. Therefore, it is possible for a pair of clusters to have no MCF links between them, which may occur if the MCF links of those clusters are distributed to other clusters.

In terms of computational complexity, the typical agglomerative clustering method using MCF linkage has a computational complexity of $O(n^3)$, which is the typical complexity of standard linkages (n is the size of the dataset). However, in the agglomerative clustering method with the iterative linkage criterion IMCF, the computational complexity increases to $O(n^4)$, due to the iterative computations involved in the IMCF case.

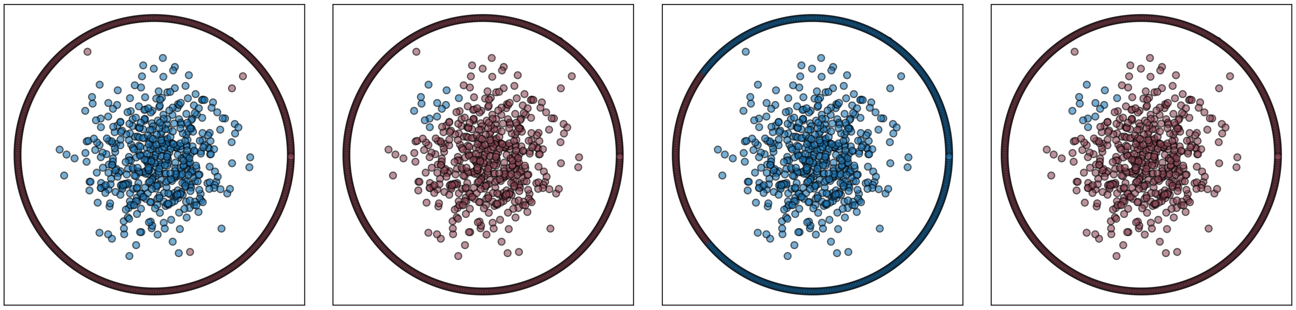
For this reason, in order to reduce execution time, in the experiments we conducted, we started with an overclustering of the dataset into a large number small clusters and then performed agglomerative cluster aggregation until the desirable number of clusters is reached.

V. EXPERIMENTS

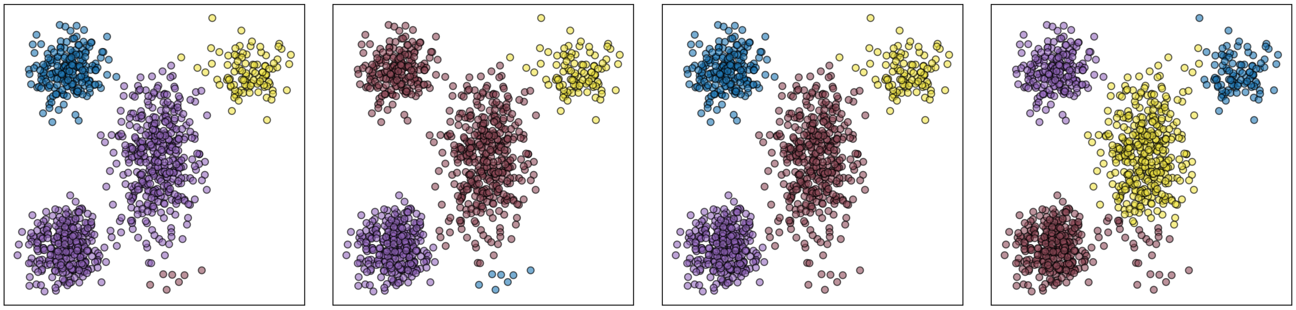
We have conducted a series of experiments using well-known synthetic and real datasets to compare the agglomera-



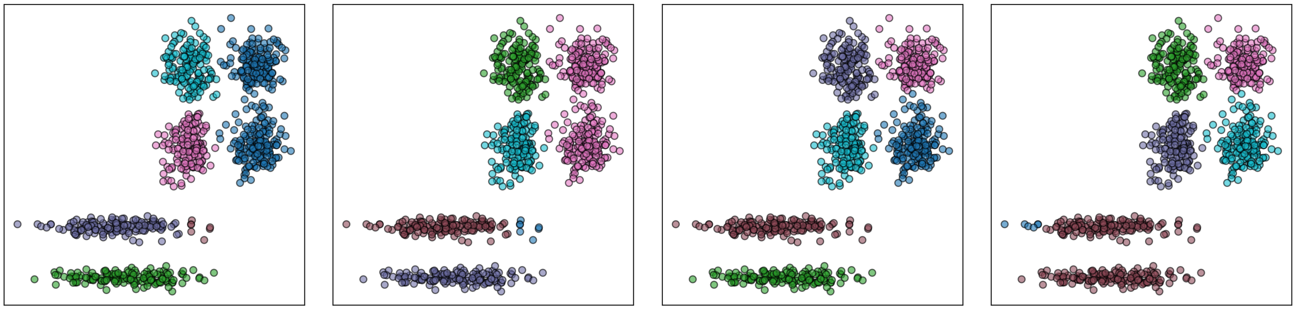
(a) MCF (b) Single (c) IMCF (d) Average



(e) MCF (f) Single (g) IMCF (h) Average



(i) MCF (j) Single (k) IMCF (l) Average



(m) MCF (n) Single (o) IMCF (p) Average

Fig. 2: Illustration of the agglomerative clustering solutions provided by the compared linkages on four synthetic datasets. Each row presents solutions from all compared linkages on the same dataset.

tive clustering performance of the proposed MCF and IMCF linkages against the conventional single and average linkages.

A. Datasets

TABLE I: The real world datasets used in the experiments. n is the number of data instances, d is the dimensionality, and k denotes the number of clusters.

Dataset	n	d	k
Digits	1797	64	10
Iris	150	4	3
Olivetti Faces	400	4096	40
Pendigits	3000	16	10
Seeds	210	7	3
Waveform-v1	5000	21	3
Wine	178	13	3

Table I summarizes the real world datasets[1] that we used for experimental evaluation, which vary in size n, dimensions d, number of clusters k.

Two datasets featuring handwritten digits have been considered. The Digits dataset consists of 1797 handwritten digit samples, where each digit is represented as an 8×8 grayscale image, covering ten classes corresponding to digits 0 through 9. The Pendigits dataset contains 10992 handwritten digit samples, from which we randomly selected 3000 samples. Each instance is represented by 16 coordinate features and belongs to one of the ten digit classes. Three datasets that feature natural measurements from plants and agricultural products have also been used in our experiments. The Iris dataset contains 150 samples of iris flowers, each described by four features related to sepal and petal dimensions, in three distinct species. The Seeds dataset contains 210 wheat kernel samples, each described by 7 geometric features, in three different varieties of wheat. The Wine dataset consists of 178 samples of wines, each characterized by 13 chemical characteristics corresponding to three different cultivars from the same region. We also included two datasets covering facial images and artificially generated waveforms. The Olivetti Faces dataset includes 400 grayscale 64×64 images of faces from 40 different individuals. The Waveform-v1 dataset consists of 5000 synthetic samples, each described by 21 continuous features, categorized into three distinct waveform classes.

In order to ensure a more thorough evaluation, our comparative experimental study also included synthetic datasets used in the literature. These datasets represent a variety of clustering challenges, including complex geometries and patterns. For example, datasets such as 2d-20c-no0, 3-spiral, and flame cover a range of difficulty levels, from well-separated clusters to intricate and overlapping structures[2].

In all datasets, we used min-max normalization as a prepossessing step, to map the features of each data point to the $[0, 1]$ interval and prevent attributes with large ranges from dominating distance calculations.

TABLE II: Clustering performance (NMI score) for real datasets.

Dataset	MCF	single	IMCF	average
Digits	0.77	0.77	**0.82**	0.80
Iris	0.72	0.72	**0.78**	0.75
Olivetti Faces	**0.68**	0.67	0.70	**0.74**
Pendigits	**0.65**	0.58	**0.72**	0.69
Seeds	0.02	0.02	**0.70**	0.55
Waveform-v1	1.00	1.00	**1.00**	0.75
Wine	0.02	0.02	**0.82**	0.02

TABLE III: Clustering performance (NMI score) for synthetic datasets.

Dataset	MCF	single	IMCF	average
2d-20c-no0	0.95	**0.97**	**0.99**	0.98
2d-4c-no4	0.67	0.67	**0.99**	0.67
3-spiral	1.00	1.00	**0.07**	0.01
DS-850	**0.85**	0.84	**0.98**	0.93
complex8	**0.86**	0.85	**0.73**	0.67
compound	0.80	0.80	**0.84**	0.81
cure-t0-2000n-2D	1.00	1.00	**1.00**	0.60
donut2	**0.90**	0.02	0.17	**0.27**
ds2c2sc13	**0.96**	0.94	**0.90**	0.82
flame	0.02	0.02	**0.94**	0.62
jain	**0.25**	0.09	0.69	0.69
longsquare	0.91	0.91	**0.98**	0.91
triangle2	0.85	0.85	**0.97**	0.89

B. Evaluation

In order to evaluate the clustering results, we used *Normalized Mutual Information* (NMI) [15] score defined as:

$$NMI(Y, C) = \frac{2 \times I(Y, C)}{H(Y) + H(C)}, \tag{7}$$

where Y denotes the ground-truth labels, C denotes the clusters labels provided the clustering method, $I(\cdot)$ is the mutual information measure and $H(\cdot)$ the entropy. NMI score takes values in $[0, 1]$ with higher values indicating better clustering performance.

For all algorithms, the number of clusters is set to the number of ground-truth categories of each dataset and we made the typical assumption that the provided class labels constitute the ground truth cluster labels.

C. Experimental Setup and Results

We compare the performance of the proposed MCF and IMCF linkages against the classical single and average linkages using the Normalized Mutual Information (NMI) metric. It should be noted that, in all cases, the agglomerative algorithm was initialized with a fine-grained clustering (overclustering) using $k_{init} = 50$ clusters to significantly reduce execution time. To obtain the initial overclustering, we employed the global k-means++ [16] due to its powerful clustering error minimization capabilities.

Tables II and III provide the results for the real and synthetic datasets, respectively. It can be observed that the MCF linkage consistently outperforms or matches the performance of a single linkage in both real-world and synthetic datasets. In particular, on the Pendigits dataset, MCF achieves a considerable improvement (NMI 0.65 vs 0.58). In the 'donut2'

---

[1] The datasets are available at: https://archive.ics.uci.edu/datasets.

[2] The synthetic datasets are available at: https://github.com/deric/clustering-benchmark.

synthetic dataset MCF achieves the best NMI significantly outperforming all other linkages. It is evident that MCF linkage can be considered as an improved alternative to single linkage.

In what concerns the IMCF linkage, more remarkable performance improvements can be observed compared to average linkage. IMCF consistently achieves the highest NMI scores in almost all real-world datasets, particularly excelling in Digits, Iris, Pendigits, Seeds, Waveform-v1 and Wine with substantial performance gaps. For example, on the Wine dataset, IMCF scores 0.82 while average linkage scores only 0.02. In synthetic datasets, IMCF outperforms average linkage in highly complex scenarios, such as 2d-4c-no4, DS-850, and flame, demonstrating its ability to successfully measure cluster separation even in the presence of complex cluster structures.

Figure 2 visually illustrates the differences in the merging behavior of the compared linkage methods on several synthetic datasets. Each row in the figure displays the agglomerative clustering solutions obtained by applying the four linkages to the same synthetic dataset.

It can be observed that MCF and single linkage demonstrate different clustering behavior. Specifically, in the first and second datasets (subfigures (a), (b) and (e), (f)), MCF better captures the clustering structure, whereas in the third and fourth datasets (subfigures (i), (j) and (m), (n)), both linkages exhibit similar behavior.

IMCF linkage achieves improved performance compared to average linkage in the third and fourth datasets, (subfigures (k), (l) and (o), (p)), demonstrating greater robustness to outliers, as these datasets contain several outlying points. Finally, in the first and second datasets (subfigures (c), (d) and (g), (h)), IMCF behaves similarly to average linkage, with both methods failing to accurately capture the underlying structure of the data.

In summary, the results of the experimental study indicate that MCF is superior to or equal to single linkage, while IMCF considerably outperforms average linkage and seems to be the preferable method in the majority of datasets.

VI. CONCLUSIONS

We have introduced new measures to quantify cluster separation based on counterfactual distances. Considering a pair of clusters, we demonstrated that the counterfactuals of points from one cluster effectively represent the frontier of the opposing cluster. As a result, the distances between the mutual counterfactual points offer a more accurate estimation of cluster separation than traditional methods.

Building on this insight, we extended standard linkage criteria by proposing the Mutual Counterfactual (MCF) and Iterative Mutual Counterfactual (IMCF) linkage criteria, which rely solely on distances between mutual counterfactuals. These linkages are versatile, as they can be applied with any distance function and require only the pairwise distance matrix. Thus, there is no need for access to the original data vectors.

We incorporated the proposed linkages into a standard agglomerative clustering framework and conducted comparative experiments against traditional single-link and average-link criteria. The experimental results demonstrate the effectiveness of using counterfactuals to assess cluster separation in more meaningful way.

Future work could focus on a more extensive experimental evaluation of the proposed linkages. Given the increased computational complexity of the IMCF linkage, developing efficient implementations by leveraging specialized data structures such as in standard agglomerative clustering would be beneficial. Additionally, it may be valuable to introduce an upper bound threshold for link distances within the IMCF criterion to control merge sensitivity. Finally, integrating these linkages with measures of cluster compactness, such as cluster densities, could lead to more sophisticated criteria for agglomerative cluster merging.

ACKNOWLEDGMENT

This 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).

REFERENCES

[1] C. K. Reddy and B. Vinzamuri, “A survey of partitional and hierarchical clustering algorithms,” in *Data clustering*. Chapman and Hall/CRC, 2018, pp. 87–110.

[2] X. Ran, Y. Xi, Y. Lu, X. Wang, and Z. Lu, “Comprehensive survey on hierarchical clustering algorithms and the recent developments,” *Artificial Intelligence Review*, vol. 56, no. 8, pp. 8219–8264, 2023.

[3] E. K. Tokuda, C. H. Comin, and L. d. F. Costa, “Revisiting agglomerative clustering,” *Physica A: Statistical mechanics and its applications*, vol. 585, p. 126433, 2022.

[4] G. Vardakas, A. Karra, E. Pitoura, and A. Likas, “Counterfactual explanations for k-means and gaussian clustering,” *arXiv preprint arXiv:2501.10234*, 2025.

[5] A. Karra, G. Vardakas, E. Pitoura, and A. Likas, “Generating counterfactual explanations for clustering models based on their equivalence to classification models,” in *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer, 2025.

[6] S. Wachter, B. Mittelstadt, and C. Russell, “Counterfactual explanations without opening the black box: Automated decisions and the gdpr,” *Harv. JL & Tech.*, vol. 31, p. 841, 2017.

[7] S. Verma, V. Boonsanong, M. Hoang, K. Hines, J. Dickerson, and C. Shah, “Counterfactual explanations and algorithmic recourses for machine learning: A review,” *ACM Computing Surveys*, vol. 56, no. 12, pp. 1–42, 2024.

[8] R. Guidotti, “Counterfactual explanations and how to find them: literature review and benchmarking,” *Data Mining and Knowledge Discovery*, vol. 38, no. 5, pp. 2770–2824, 2024.

[9] A. Dhurandhar, P.-Y. Chen, R. Luss, C.-C. Tu, P. Ting, K. Shanmugam, and P. Das, “Explanations based on the missing: Towards contrastive explanations with pertinent negatives,” *Advances in neural information processing systems*, vol. 31, 2018.

[10] T. Le, S. Wang, and D. Lee, “Grace: Generating concise and informative contrastive sample to explain neural network model’s prediction,” in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 238–248.

[11] H.-G. Jung, S.-H. Kang, H.-D. Kim, D.-O. Won, and S.-W. Lee, “Counterfactual explanation based on gradual construction for deep networks,” *Pattern Recognition*, vol. 132, p. 108958, 2022.

[12] R. K. Mothilal, A. Sharma, and C. Tan, “Explaining machine learning classifiers through diverse counterfactual explanations,” in *Proceedings of the 2020 conference on fairness, accountability, and transparency*, 2020, pp. 607–617.

[13] S. Sharma, J. Henderson, and J. Ghosh, "Certifai: Counterfactual explanations for robustness, transparency, interpretability, and fairness of artificial intelligence models," *arXiv preprint arXiv:1905.07857*, 2019.

[14] M. Schleich, Z. Geng, Y. Zhang, and D. Suciu, "Geco: Quality counterfactual explanations in real time," *arXiv preprint arXiv:2101.01292*, 2021.

[15] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE Transactions on neural networks*, vol. 20, no. 2, pp. 189–201, 2009.

[16] G. Vardakas and A. Likas, "Global k-means++: an effective relaxation of the global k-means clustering algorithm," *Applied Intelligence*, vol. 54, no. 19, pp. 8876–8888, 2024.