Counterfactual Explanations for k-means and Gaussian Clustering

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*—Counterfactuals have been recognized as an effective approach to explain classifier decisions. In this work, we focus on the use of counterfactuals to explain clustering solutions. First, we present a general definition for counterfactuals for model-based clustering that includes plausibility and feasibility constraints. Then we consider the counterfactual generation problem for k-means and Gaussian clustering assuming Euclidean distance. Our approach takes as input the factual, the target cluster, a binary mask indicating actionable or immutable features and a plausibility factor specifying how far from the cluster boundary the counterfactual should be placed. In the k-means clustering case, analytical mathematical formulas are presented for computing the optimal solution, while in the Gaussian clustering case (assuming full, diagonal, or spherical covariances) our method requires the numerical solution of a nonlinear equation with a single parameter only. We demonstrate the advantages of our approach through illustrative examples and quantitative experimental comparisons.**

***Index Terms*—Explainable AI, counterfactuals, clustering, k-means, Gaussian clusters.**

I. Introduction

Explainable AI is essential for building trustworthy and transparent machine learning models, allowing users to understand, evaluate, and effectively interact with these systems. Explanation methods can be broadly categorized into global and local methods [1]. Global methods aim to provide insights into the overall behavior of a model, often using decision trees or rule-based systems to summarize patterns across the entire dataset. In contrast, local methods focus on individual predictions, offering insights into specific outcomes.

Counterfactual explanations (CFEs) are a well-established *local explanation* method for explaining decisions of machine learning models [2], [3]. The widely cited 'Alice' example for loan qualification illustrates the core idea: Suppose that the model f has two possible outcomes: $c =$ `'negative'`, meaning the applicant does not qualify for the loan, and $c' =$ `'positive'`, meaning the applicant does qualify. Say Alice is denied the loan because $f(y) = c$, where y is her input vector. An explanation is needed to help Alice get the loan in the future: what is the minimum required change that Alice should make (in terms of income, education, etc.) to qualify for the loan. This is the kind of explanation counterfactuals offer.

In general, given a classifier f that outputs the decision $c = f(y)$ for a factual instance y, a counterfactual explanation is an instance z such that (i) the decision for f on z is different from c, i.e., $f(z) \neq c$, and (ii) the distance between y and z is minimum [2], [3]. Several interesting properties of counterfactuals for classification have been defined, such as *actionability* (some features are not allowed to change) and *plausibility* (the counterfactual should lie inside the data manifold).

Although substantial research has addressed counterfactual explanations for classification, counterfactuals for clustering have received limited attention. This paper seeks to bridge this gap through the following contributions:

- We propose a general definition of counterfactuals for clustering assuming that each cluster is represented by a probability density. This formulation subsumes k-means as a special case.
- We introduce efficient, non-iterative methods for generating counterfactuals under *Euclidean distances* between factual and counterfactual instances. For k-means clustering, we derive closed-form analytical formulas; for Gaussian clustering, we formulate solvable equations for full, diagonal and spherical covariance structures.
- We extend our approach to account for: (i) actionable and immutable features and (ii) counterfactual plausibility by moving from the cluster boundary towards regions of higher cluster density.

To assess the effectiveness of our approach, we use several UCI datasets. We compare the generated counterfactuals to those provided by considering an indirect approach that treats the problem as classification and exploits available tools and methods.

The rest of the paper is organized as follows. Section II briefly discusses related work. In Section III we present our counterfactual definition followed by the general framework for counterfactual computation in Section IV. In Section V we derive the analytical formulas for the k-means case, while in Section VI the equations for the case of Gaussian clusters. In Section VII we present experimental results, while Section VIII provides conclusions and future research directions.

II. Related Work

Explainable methods for clustering focus mostly on *global explanations*. In particular, they build *decision tree* models that directly provide interpretations in the form of decision rules. Several methods have been proposed to build decision trees to explain clustering by utilizing indirect or direct approaches.

The *indirect* global explanation methods, typically follow a two-step procedure: first, cluster labels are obtained using a clustering algorithm, such as k-means [4], and then a supervised decision tree algorithm is applied to build a decision tree that interprets the resulting clusters [5]. *Direct* global explanation methods integrate decision tree construction and partitioning into clusters. Many of them follow the typical top-down splitting procedure used in the supervised case but exploit unsupervised splitting criteria, e.g., compactness of the resulting subsets [6] or unimodality criteria [7].

The *local explanation* problem has received limited attention. In this paper, we focus on counterfactual-based local explanations for clustering. Counterfactual explanations are local explanations proposed for classification [2], [3], [8]. We present a novel general definition for counterfactual in clustering and introduce efficient, non iterative counterfactual generation approaches for k-means and Gaussian clustering. Local explanations enable a deeper understanding of individual assignments and their relationship to cluster characteristics. Furthermore, in addition to explaining the cluster assignment of a specific instance, counterfactual explanations reveal the minimum feature changes needed to assign an instance to a different cluster.

In what concerns counterfactuals for clustering, in [9] a specific scoring function defined for classification is modified for the clustering problem and optimized using Bayesian approaches. In [10] the authors formulate the clustering problem as a classification problem and use known methods from classification to generate counterfactuals.

III. Counterfactuals For Model-Based Clustering

We assume the general clustering problem with $C_1, ..., C_M$ clusters. We are given the probability density $p_k(x)$ for each cluster C_k, $k = 1, ..., M$. Let also $(\pi_1, ..., \pi_M)$ be the prior probability vector of the clusters. Typically π_k is set equal to the cluster frequencies ($\pi_k = N_k/N$) or equal to $1/M$. Based on the *cluster assignment rule*, an example x is assigned to cluster C_ℓ for which $\pi_\ell p_\ell(x)$ is maximum.

In order to compute a counterfactual explanation (CFE) z for a given example (factual) y, we need to specify a *preference density* $r(x|y)$ which should be a *unimodal distribution* with mode at y expressing the *preference* for x to become a counterfactual of y. This preference density definition is very general and flexible and also implements feature actionability, as will be discussed later.

CFE Definition: Given a factual y of cluster C_k, its *counterfactual explanation* (CFE) z is defined as the solution to the following constrained optimization problem:

$$z = \arg\max_x r(x|y) \text{ (z has maximum preference given y)} \tag{1}$$

subject to the constraint:

$$C_\ell \neq C_k \text{ where } C_\ell = \arg\max_{C_m} \pi_m p_m(z) \tag{2}$$

In the above formulation, C_ℓ is the cluster label of z, taking into account the cluster assignment rule. The preference density $r(x|y)$ should be a unimodal function (with peak at the factual y) that should decrease as the distance $d(x, y)$ between x and y increases. The exponential form ($r(x|y) = exp(-d(x, y))$) is a viable option, although other functional forms could be devised. Note that, if $r(x|y)$ is monotonically decreasing with respect to $d(x, y)$, then the maximization of $r(x|y)$ is equivalent to the minimization of $d(x, y)$.

It is possible to extend the above set of constraints. More specifically a *plausibility constraint* can be specified on the counterfactual z:

$$p_\ell(z) > \delta \text{ (plausibility)} \tag{3}$$

where the parameter $\delta > 0$ is a threshold on the local cluster density of z. Adjusting δ allows the counterfactual to lie in regions of sufficient cluster density, i.e., it does not constitute an outlier.

Moreover, by appropriately defining $r(x|y)$, we can introduce *feasibility* (or actionability) constraints:

$$r(z|y) > 0 \text{ (feasibility)} \tag{4}$$

i.e., impose constraints on the allowed feature values of counterfactual z. If for example the values of the i-th feature are not allowed to change, we can set $r(x|y) = 0$ for all x with $x_i \neq y_i$. In this case, the feasibility constraint will prevent the generation of counterfactual z with $z_i \neq y_i$.

In the following, we focus on *Gaussian clusters*, i.e. the probability density of each cluster i follows the Gaussian distribution $p_i(x) = N(x; \mu_i, S_i)$ where μ_i is the mean (cluster center) and S_i is the covariance matrix of the distribution. It should be stressed that k-means clustering is included in this framework since it can be considered a special case of Gaussian clustering. A convenient property of a Gaussian cluster is that its density $p(x)$ is inversely proportional to the (Mahalanobis in the general case) distance of the cluster center. Therefore, considering two clusters i and j, there exists a cluster boundary $B(i, j)$, i.e. a set of points x for which $\pi_i p_i(x) = \pi_j p_j(x)$. Assuming that the cluster centers do not coincide, as we depart from the cluster boundary and move inside cluster j, the density p_j increases, while the density p_i decreases. Note also that cluster assignment is based on the maximum cluster density. Consequently, given a factual point y of cluster i (i.e. $\pi_i p_i(y) > \pi_j p_j(y)$), it is ensured that its closest counterfactual z (with respect to target cluster j) will be located on the cluster boundary, i.e., $z \in B(i, j)$. This fact constitutes the basis of our methods. Given a factual y, the

counterfactual z is computed as its closest point that lies on the cluster boundary. This idea is then extended to address the case where the counterfactual is desired to be positioned not at the cluster boundary, but within a more dense region of the target cluster, in order to achieve higher plausibility. In this case, the user should specify a parameter that we call *plausibility factor*.

IV. Counterfactual Computation: General Framework

In all algorithms for counterfactual generation that are presented below, we consider *pairs of clusters*: *with C_s denoting the source cluster of factual y and C_t the target cluster of counterfactual z*. The reason is that, if more than one counterfactual clusters exist (e.g. C_2, C_3 etc), we can apply the algorithms separately for each cluster pair (C_s, C_t) $(t \neq s)$, compute the corresponding counterfactuals and keep the one with maximum preference (e.g. minimum distance from the factual).

The clustering information that should be given is the clustering model, i.e. the two cluster centers in the k-means case or the parameters of the two Gaussians and the priors π in the Gaussian clustering case. *No data availability is required*.

We assume a preference function of the form:

$$r(x|y) = \exp(-d(x,y)) \tag{5}$$

where $d(x,y) = |x-y|^2$ is the *squared Euclidean distance*. Thus, maximizing $r(x|y)$ in eq. 1 is equivalent to *minimizing Euclidean distance*.

Our method takes as input the factual y and the cluster model parameters. In order to account for plausibility, the user might specify a *plausibility factor* ϵ to indicate whether the counterfactual should be located at the cluster boundary ($\epsilon = 0$) or it should depart from the cluster boundary being placed inside the target cluster region for increasing values of $\epsilon > 0$. The cluster model along with the plausibility factor value ϵ define a *constraint equation* that determines the set of points where the counterfactual should be located. We call this set *constraint set* CS_ϵ. If $\epsilon = 0$, the constraint set is identical to the boundary B between the two clusters.

Our method also accounts for actionable and immutable features, i.e., which features are allowed to change and which are not. This is implemented by a user-defined binary mask vector M that indicates which components of the parameter vector z are allowed to change with respect to the factual y. If $M_i = 1$, z_i is considered as free (actionable) parameter allowed to change. If $M_i = 0$ then $z_i = y_i$, i.e. z_i remains fixed (immutable).

Given the factual y, the plausibility factor ϵ and the feature mask M, we present below how to compute counterfactuals in the case of k-means clustering as well as for Gaussian clusters with full, diagonal and spherical covariances. The solution is analytical in the k-means case and non-iterative, requiring the solution of a nonlinear system with a single parameter, in the Gaussian case.

V. Counterfactuals For k-means Clustering

Let m_s and m_t be two cluster centers. Based on the k-means cluster assignment rule, a point z is assigned to the target cluster if $|z-m_t|^2 < |z-m_s|^2$. Therefore, the cluster boundary contains those points z for which $|z-m_s|^2 = |z-m_t|^2$. In order to account for plausibility, we define the constraint set CS_ϵ of candidate counterfactual locations, using the following constraint equation:

$$CS_\epsilon = \{z : |z-m_s|^2 = |z-m_t|^2 + \epsilon|m_t - m_s|^2\}, \epsilon \geq 0 \tag{6}$$

For $\epsilon = 0$ the cluster boundary is defined.

We are given the factual y and a binary mask M indicating the elements of z that are allowed to change during optimization. We split z based on the free and fixed components: we denote by z_f the subvector of free components of z (with $M_i = 1$) and as z_{fixed} the subvector of fixed components of z (with $M_i = 0$), hence $z_{fixed} = y_{fixed}$. We wish to find the vector $z \in CS_\epsilon$ of the above form with minimum squared Euclidean distance from factual y.

Let $d_\epsilon = \epsilon|m_t - m_s|^2$. The constraint equation for the set CS_ϵ can be written as

$$z^\top (m_s - m_t) = \frac{|m_s|^2 - |m_t|^2 - d_\epsilon}{2}. \tag{7}$$

Let $v = m_s - m_t$ and $c = \frac{|m_s|^2 - |m_t|^2 - d_\epsilon}{2}$. Then the above constraint is written as $z^\top v = c$.

Let also v_f and v_{fixed} be the subvectors of v with free and fixed components, respectively. The linear constraint equation can be written as:

$$z_f^\top v_f + y_{fixed}^\top v_{fixed} = c. \tag{8}$$

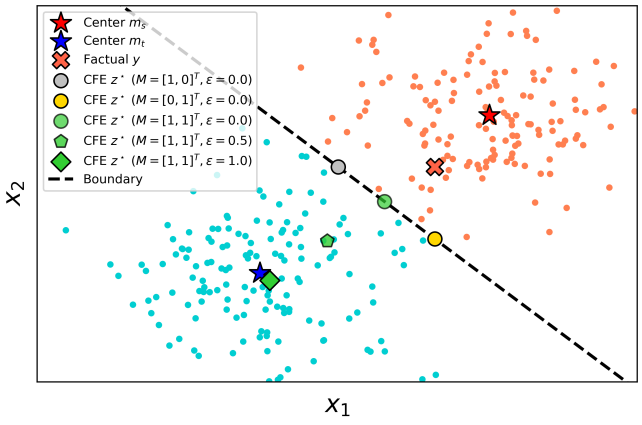


Fig. 1: Illustration of the counterfactuals computed in the case of a 2-d synthetic dataset partitioned in two clusters using k-means. The factual y is the same and the counterfactuals computed for several values of mask M and plausibility factor ϵ are presented.

The optimization problem with parameter vector z_f is defined as follows:

$$\text{minimize } |z_f - y_f|^2$$

subject to

$$z_f^\top v_f = c - y_{fixed}^\top v_{fixed}.$$

This is a quadratic minimization problem with a linear constraint, which can be solved either using Lagrange multipliers or by projecting y_f onto the hyperplane defined by this constraint. The latter approach is straightforward and gives the solution for the free part of z:

$$z_f^\star = y_f - \frac{(y_f^\top v_f - (c - y_{fixed}^\top v_{fixed}))}{|v_f|^2} v_f. \tag{9}$$

The fixed components of $z^\star$ will be set equal to the corresponding y_i values, i.e. $z^\star_{fixed} = y_{fixed}$.

It should be stressed that *the above solution is valid only if* $v_f \neq 0$ or $y_{fixed}^\top v_{fixed} \neq c$. In the opposite case, a valid counterfactual does not exist.

Fig. 1 provides a visual illustration of the optimal counterfactuals computed for a 2-d synthetic dataset partitioned in two clusters using k-means. Varying the plausibility factor ϵ moves the counterfactual from the cluster boundary into the target cluster region. The actionability mask M constrains changes: with $M = [1, 0]$ (or $[0, 1]$), only the horizontal (or vertical) coordinate of the factual can change.

VI. Counterfactuals For Gaussian Clusters

A. Gaussian clusters with full covariance

Let s and t be the source and target clusters with densities $p_s(x) = \pi_s N(x; m_s, S_s)$ and $p_t(x) = \pi_t N(x; m_t, S_t)$ where m_t, m_t are the mean vectors, S_s, S_t are full covariance matrices and π_s, π_t the cluster priors.

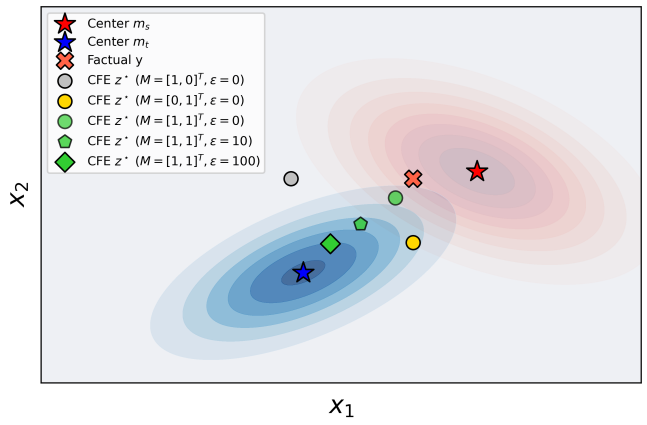


Fig. 2: Illustration of the counterfactuals computed in the case of two Gaussian clusters with full covariances. The factual y is the same and the counterfactuals computed for several values of mask M and plausibility factor ϵ are presented.

In probabilistic clustering, a point z is assigned to the cluster of maximum density. Thus, the cluster boundary contains the points z with $p_s(z) = p_t(z)$. In order to account for plausibility, we define the constraint set CS_ϵ of possible counterfactual location, using the following constraint equation:

$$CS_\epsilon = \{z : p_t(z) = (1 + \epsilon) p_s(z)\}, \epsilon \geq 0 \tag{10}$$

For $\epsilon = 0$ the cluster boundary is defined.

We are given the factual y, the mask vector M and the plausibility factor ϵ and our goal is to find $z \in CS_\epsilon$ that minimizes $|z - y|^2$ taking into account the mask M. Based on mask M, we define the index sets:

- Free indices: $F = \{i : M_i = 1\}$.
- Fixed indices: $G = \{i : M_i = 0\}$.

Therefore, we can group the indices of z and m_k as follows:

- $z = [z_F, z_G]^\top$, where $z_G = y_G$ (fixed).
- $m_k = [m_{kF}, m_{kG}]^\top$, for $k \in \{s, t\}$.

Moreover, the covariance inverses S_k^{-1}, $k \in \{s, t\}$ can also be partitioned into blocks:

$$S_k^{-1} = \begin{bmatrix} S_k^{-1,FF} & S_k^{-1,FG} \\ S_k^{-1,GF} & S_k^{-1,GG} \end{bmatrix}$$

The optimization problem to be solved is defined:

$$\text{minimize} \sum_{i \in F} (z_i - y_i)^2 \tag{11}$$

subject to the constraint:

$$\begin{aligned} &(z - m_t)^\top S_t^{-1} (z - m_t) - (z - m_s)^\top S_s^{-1} (z - m_s) \\ &+ \ln \frac{|S_t|}{|S_s|} - 2 \ln \frac{\pi_t}{\pi_s} + 2 \ln(1 + \epsilon) = 0 \end{aligned} \tag{12}$$

where the above equation is obtained by taking the logarithm of both sides in the constraint equation 10. To simplify notation we define the constant

$$c_\alpha = \ln \frac{|S_t|}{|S_s|} - 2 \ln \frac{\pi_t}{\pi_s} + 2 \ln(1 + \epsilon) \tag{13}$$

and replace in the above constraint equation.

The above quadratic optimization problem with a quadratic equality constraint can be solved by introducing a Lagrange multiplier λ and defining the Lagrangian function:

$$\begin{aligned} \mathcal{L}(z_F, \lambda) = \sum_{i \in F} (z_i - y_i)^2 - \lambda \Big[&(z - m_t)^\top S_t^{-1} (z - m_t) \\ &- (z - m_s)^\top S_s^{-1} (z - m_s) + c_\alpha \Big] \end{aligned} \tag{14}$$

Taking the gradient of L with respect to z_F equal to zero:

$$z_F = (I - \lambda D)^{-1} (y_F - \lambda d) \tag{15}$$

where

$$D = S_t^{-1,FF} - S_s^{-1,FF} \tag{16}$$

$$\begin{aligned} d = &S_t^{-1,FF} m_{tF} - S_s^{-1,FF} m_{sF} \\ &- \left(S_t^{-1,FG} (z_G - m_{tG}) - S_s^{-1,FG} (z_G - m_{sG}) \right) \end{aligned} \tag{17}$$

Setting the gradient of the constraint equation with respect to λ equal to zero and substituting z from Eq. 15 we get the non-linear equation to be solved for λ :

$$\begin{aligned} &(y_F - \lambda d)^\top (I - \lambda D)^{-1} D (I - \lambda D)^{-1} (y_F - \lambda d) \\ &- 2 (y_F - \lambda d)^\top (I - \lambda D)^{-1} e + c_f + c_\ell + c_g + c_\alpha = 0 \end{aligned} \tag{18}$$

where

$$e = S_t^{-1,FF} m_{tF} - S_s^{-1,FF} m_{sF} \tag{19}$$

$$c_f = m_{tF}^\top S_t^{-1,FF} m_{tF} - m_{sF}^\top S_s^{-1,FF} m_{sF} \tag{20}$$

$$\begin{aligned} c_g = &(z_G - m_{tG})^\top S_t^{-1,GG} (z_G - m_{tG}) \\ &- (z_G - m_{sG})^\top S_s^{-1,GG} (z_G - m_{sG}) \end{aligned} \tag{21}$$

$$\begin{aligned} c_\ell = &2(z_F - m_{tF})^\top S_t^{-1,FG} (z_G - m_{tG}) \\ &- 2(z_F - m_{sF})^\top S_s^{-1,FG} (z_G - m_{sG}) \end{aligned} \tag{22}$$

Once the solution $\lambda^\star$ is found, it is substituted in Eq. 15 to provide the counterfactual $z_F^\star$. For the fixed parameters, $z_G^\star = y_G$. For the uniqueness of the solution, if matrix $S_t^{-1,FF} - S_s^{-1,FF}$ is positive or negative definite then a unique solution exists. Otherwise, there is the possibility for one, multiple or no solutions. Fig. 2 shows the generated counterfactuals for two Gaussian clusters with full covariances. The boundary is quadratic in this case. As ϵ increases, counterfactuals move deeper into the target cluster. The actionability mask M constrains movement: $M = [1, 0]$ limits changes to the horizontal axis, while $M = [0, 1]$ to the vertical.

B. Gaussian clusters with diagonal covariance

Let $p_s(x) = \pi_s N(x; m_s, S_s)$ and $p_t(x) = \pi_t N(x; m_t, S_t)$ where S_s and S_t are *diagonal* covariance matrices, i.e.

$$S_s = \text{diag}(\sigma_{s1}^2, \ldots, \sigma_{sd}^2) \text{ and } S_t = \text{diag}(\sigma_{t1}^2, \ldots, \sigma_{td}^2).$$

The constraint $p_t(z) = (1 + \epsilon) \cdot p_s(z)$ translates to:

$$\begin{aligned} &\sum_{i \in F} \left(\frac{(z_i - m_{ti})^2}{\sigma_{ti}^2} - \frac{(z_i - m_{si})^2}{\sigma_{si}^2} \right) + \\ &\sum_{i \in G} \left(\frac{(y_i - m_{ti})^2}{\sigma_{ti}^2} - \frac{(y_i - m_{si})^2}{\sigma_{si}^2} \right) + c_\alpha = 0 \end{aligned} \tag{23}$$

where c_α is given by Eq. 13 with $|S_k| = \prod_{j=1}^d \sigma_{kj}^2$, $k \in \{s, t\}$.

Our objective is to minimize $\sum_{i \in F} (z_i - y_i)^2$ subject to the constraint equation 23. We define the Lagrangian:

$$\begin{aligned} &\mathcal{L}(z, \lambda) = \sum_{i \in F} (z_i - y_i)^2 + \\ &\lambda \left(\sum_{i \in F} \left(\frac{(z_i - m_{ti})^2}{\sigma_{ti}^2} - \frac{(z_i - m_{si})^2}{\sigma_{si}^2} \right) + c_h + c_\alpha \right) \end{aligned} \tag{24}$$

where $c_h = \sum_{i \in G} \left(\frac{(y_i - m_{ti})^2}{\sigma_{ti}^2} - \frac{(y_i - m_{si})^2}{\sigma_{si}^2} \right)$ is a constant. For $i \in F$, taking the partial derivative of $\mathcal{L}$ with respect to z_i and setting to zero, we get:

$$z_i = \frac{y_i + \lambda \left(\frac{m_{si}}{\sigma_{si}^2} - \frac{m_{ti}}{\sigma_{ti}^2} \right)}{1 + \lambda \left(\frac{1}{\sigma_{ti}^2} - \frac{1}{\sigma_{si}^2} \right)}. \tag{25}$$

Setting the gradient of the constraint equation with respect to λ equal to zero and substituting z from Eq. 25 we get the non-linear equation that is numerically solved for λ :

$$\begin{aligned} &\sum_{i \in F} \frac{\left(\frac{y_i - \lambda \left(\frac{m_{si}}{\sigma_{si}^2} - \frac{m_{ti}}{\sigma_{ti}^2} \right)}{1 + \lambda \left(\frac{1}{\sigma_{si}^2} - \frac{1}{\sigma_{ti}^2} \right)} - m_{ti} \right)^2}{\sigma_{si}^2} - \\ &\sum_{i \in F} \frac{\left(\frac{y_i + \lambda \left(\frac{m_{si}}{\sigma_{si}^2} - \frac{m_{ti}}{\sigma_{ti}^2} \right)}{1 + \lambda \left(\frac{1}{\sigma_{si}^2} - \frac{1}{\sigma_{ti}^2} \right)} - m_{si} \right)^2}{\sigma_{ti}^2} + c_h + c_\alpha = 0. \end{aligned} \tag{26}$$

If a solution $\lambda^\star$ is found, then each free z_i can be computed by substituting $\lambda^\star$ in Eq. 25. The fixed elements are simply $z_i^\star = y_i$ for $i \in G$. In what concerns the uniqueness of the solution, it is guaranteed in the case where the sign of $\frac{1}{\sigma_{si}^2} - \frac{1}{\sigma_{ti}^2}$ is the same for all free dimensions $i \in F$. Otherwise, there is the possibility for one, multiple or no solutions.

C. Gaussian clusters with spherical covariance

Let $p_s(x) = \pi_s N(x; m_s, S_s)$ and $p_t(x) = \pi_t N(x; m_t, S_t)$ where S_s and S_t are spherical covariance matrices: $S_s = \sigma_s^2 I$ and $S_t = \sigma_t^2 I$. The constraint $p_t(z) = (1 + \epsilon) \cdot p_s(z)$ translates to:

$$\frac{|z - m_t|^2}{\sigma_t^2} - \frac{|z - m_s|^2}{\sigma_s^2} + c_\alpha = 0. \tag{27}$$

where c_α is given by Eq. 13 with $|S_k| = \sigma_k^{2d}$, $k \in \{s, t\}$. The objective is to minimize $\sum_{i \in F} (z_i - y_i)^2$, subject to the constraint of Eq. 27

By replacing $\sigma_{sj}^2 = \sigma_s^2$ and $\sigma_{tj}^2 = \sigma_t^2$ for all j in Eq. 25 and Eq. 26, the equations for z_i and λ are directly obtained. It should be noted that a unique solution exists in this case.

VII. Experimental Results

To demonstrate the effectiveness of our approach, Counterfactuals for Clustering (CFCLUST), we first present two illustrative examples: one using digit images and another using the Iris dataset. We then present performance results, evaluating both optimization capabilities and runtime, across several UCI datasets[1]. When using our method for Gaussian clustering, to solve for λ, we use the root finding method of the SciPy library[2].

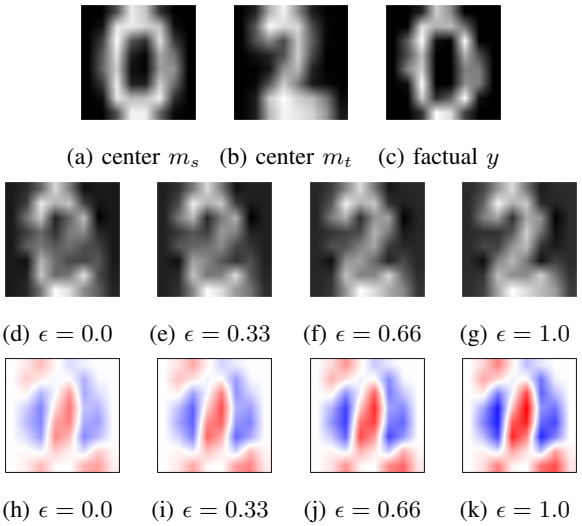
A. Illustrative examples

We used the well-studied UCI OptDigits, a 10-class dataset that contains grayscale images of handwritten digits. Each digit image is represented by a 64-dimensional feature vector that contains the intensity values of a 8×8 grid of pixels. We ignored the class labels and applied k-means to partition the dataset into ten clusters. Then we designated the cluster with the majority of images representing the digit zero as the source cluster C_s and the cluster with the majority of images representing the digit two as the target cluster C_t.

[1] https://archive.ics.uci.edu/

[2] https://docs.scipy.org/doc/scipy/reference/optimize.root-hybr.html

The top row of Fig. 3 presents the images of the centers m_s and m_t of the two clusters and the image of the factual instance y that belongs to the source cluster. Given the vectors m_s, m_t and y, we used Eq. 9, assuming no immutable features, to compute counterfactuals for increasing values of the plausibility factor ϵ. The images of the generated counterfactuals are presented in the middle row. It can be observed that when $\epsilon = 0$, the shape of counterfactual digit is actually a blend of zero and two, since the counterfactual lies on the cluster boundary. As the value of ϵ increases, the shape of the counterfactual digit clearly corresponds to digit two. Pixel-wise difference visualizations presented in the third row highlight the minimal changes needed. Red colors indicate increase in intensity, while blue colors indicate decrease in intensity. It should be noted that pixels lying near the vertical borders are mostly untouched.



(a) center m_s (b) center m_t (c) factual y

(d) $\epsilon = 0.0$ (e) $\epsilon = 0.33$ (f) $\epsilon = 0.66$ (g) $\epsilon = 1.0$

(h) $\epsilon = 0.0$ (i) $\epsilon = 0.33$ (j) $\epsilon = 0.66$ (k) $\epsilon = 1.0$

Fig. 3: Illustrative example with images of the OptDigits dataset. Top row shows images of cluster centers and the factual image; middle row shows counterfactual images for increasing values of ϵ; bottom row visualizes pixel-wise differences (blue: negative, red: positive).

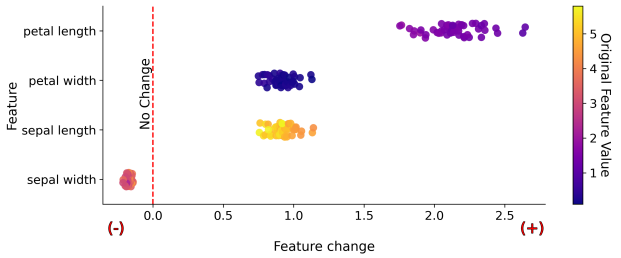


Fig. 4: Feature changes for the transition between two clusters in the Iris dataset. X-axis shows value changes; point colors reflect original feature values. Red dashed line marks the point of no change.

Additionally, we used the UCI Iris dataset, containing 150 samples described by four flower measurements. Ignoring class labels, we applied k-means to cluster the data into three groups. The cluster dominated by Iris Setosa samples was selected as the source cluster C_s, and the one dominated by Iris Virginica as the target cluster C_t.

In Fig. 4, we present a visualization showing the minimal feature changes required for an Iris sample from the source cluster to be classified into the target cluster. The plot reveals that the target cluster generally has larger petal dimensions (both length and width) and greater sepal length. In contrast, the sepal width remains similar between the clusters, as indicated by the small deviation from the red dashed line.

B. Empirical Evaluation

Experimental Setup. We evaluate our method using five well-known UCI datasets (see first column of Table I) of varying size and dimension. For each dataset we first apply clustering using either k-means or by training a Gaussian mixture model with full covariance. Then we select a source cluster C_s and a target cluster C_t and proceed with the evaluation of the counterfactual generation methods.

We first build a binary classifier with class labels corresponding to the source and target clusters [10]. Given an example y in C_s, we apply a known counterfactual generation algorithm for classification to produce a counterfactual z. To avoid classification errors, instead of using the cluster labels to train the classifier, in the case of the k-means clustering, we use a *Logistic Regression* classifier (LR) whose decision hyperplane coincides with the linear cluster boundary defined by the cluster centers m_s and m_t. In the Gaussian clustering case, the cluster boundary is quadratic and we specify as our classifier, a *Quadratic Discriminant Analysis* (QDA) whose parameters are determined by the means, covariances and priors of the Gaussian clusters.

For generating counterfactuals in the above classification framework, we employ two state-of-the-art algorithms, namely DiCE (DICE) [11] and GuidedByPrototypes (PRT) [12]. For DICE, we use the implementation provided by the authors[3], while for PRT, we use the implementation in Alibi Explain[4]. Their parameters determined after careful tuning. We set $\epsilon = 10^{-5}$ to generate solutions nearly on the cluster boundary.

Experimental Results. We report results across all datasets, considering both fully actionable features and cases with immutable features. For each setting, counterfactuals are generated for 50 randomly selected factuals from the source cluster. Unlike CFCLUST, DICE and PRT do not always produce instances belonging to the target cluster. Distance comparisons include only factuals for which all methods were successful.

In Table I we provide a comprehensive comparison of CFCLUST with PRT and DICE for both k-means and Gaussian clustering. For each dataset, we present the average squared L_2 distance between factual and counterfactual instances (dist), the result of a statistical significance test (t-test), and the

[3]https://github.com/interpretml/DiCE

[4]https://docs.seldon.io/projects/alibi/en/stable/

TABLE I: Comparison of CFCLUST with PRT and DICE for k-means and Gaussian clustering in terms of: average squared L_2 distances ('dist'), t-test significance of CFCLUST versus PRT, and DICE, and average runtimes for PRT and DICE (CFCLUST always runs in less than 0.001s, so not shown). In the 'Dataset' column, # denotes the number of immutable features. † and ‡ mark failures due to excessive time or method limitations. Best results are presented in bold.

	k-means clustering							Gaussian clustering						
	CFCLUST	PRT			DICE			CFCLUST	PRT			DICE		
Dataset	dist	dist	t-test	time	dist	t-test	time	dist	dist	t-test	time	dist	t-test	time
Iris	**0.40**	0.60	+	149.43	0.60	+	0.04	**0.40**	0.71	+	165.32	0.63	+	0.04
Iris #1	**0.41**	0.42	+	178.99	0.60	+	0.05	**0.23**	0.71	+	218.66	0.51	+	0.04
Iris #2	0.44	**0.43**	=	183.39	0.54	+	0.05	**0.41**	0.68	+	218.66	0.60	+	0.07
Wine	**0.41**	1.66	+	159.97	2.82	+	0.05	**1.18**	2.55	+	80.47	4.02	+	0.05
Wine #4	**0.41**	0.81	+	263.57	1.63	+	0.12	**1.20**	2.60	+	79.83	5.74	+	0.12
Wine #7	**0.77**	0.99	+	84.61	1.31	+	0.04	**3.01**	4.88	+	93.69	4.61	+	0.08
Pendigits	**0.94**	1.15	+	97.43	1.90	+	0.12	**0.42**	1.24	+	78.50	1.63	+	0.08
Pendigits #3	**0.96**	1.09	+	129.27	1.56	+	0.38	**0.82**	1.29	+	138.16	1.55	+	0.37
Pendigits #6	**1.16**	1.46	+	143.53	1.59	+	0.71	**0.53**	2.59	+	132.62	1.04	+	0.21
Mice-protein	**0.55**	0.98	+	79.34	1.89	+	0.29	**0.13**	1.01	+	126.58	2.19		0.59
Mice-protein #35	**0.55**	1.00	+	88.32	1.86	+	0.29	**0.51**	‡	‡	‡	†	†	†
Waveform	**0.44**	0.71	+	93.56	1.31	+	0.09	**0.25**	0.66	+	76.2	1.21	+	0.09
Waveform #11	**0.42**	0.71	+	96.41	1.06	+	19.12	**0.30**	0.67	+	77.61	0.87	+	0.14

average runtime in seconds (time). The significance level for the t-test is set at $\alpha = 0.05$, with symbol '+' indicating that CFCLUST significantly outperforms the competing method, and '=' denoting no significant difference. As it can be observed, CFCLUST consistently achieves lower average distances than both PRT and DICE, often with substantial margins, while maintaining negligible runtime (in all cases less than 0.001 seconds). This superiority is statistically significant in nearly all settings, particularly in higher-dimensional or more complex datasets where the optimization challenges are greater. Even in lower-dimensional cases, CFCLUST is superior. PRT and DICE exhibit higher computational costs and in some settings they fail due to either timeouts (†) or methodological limitations (‡), further emphasizing the robustness and efficiency of CFCLUST. In summary, it is evident that the proposed CFCLUST provides superior solutions at negligible computation time without requiring any parameter tuning.

VIII. Conclusions

In this work we have considered the use of counterfactuals to explain *clustering* solutions. At first, we have presented a general definition for counterfactuals for clustering assuming that each cluster is modeled using a probability density. Then we considered the counterfactual generation problem for k-means and Gaussian clustering assuming squared Euclidean distance among the factual and the counterfactual. Our method generates the counterfactual z through simple computational procedures, using analytical or non-iterative solutions, unlike other methods that rely on iterative optimization, often requiring parameter tuning and yielding suboptimal results. Our experiments clearly demonstrate its optimality and ease of use.

Future work could explore alternative distance measures (e.g., L_1 norm solved with iterative search), kernel-based clustering, and generating multiple diverse counterfactuals. Given its efficiency, the method could also be used to study cluster separability and fairness.

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] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD conference*, 2016, pp. 1135–1144.

[2] 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.

[3] 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.

[4] 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.

[5] E. Laber, L. Murtinho, and F. Oliveira, "Shallow decision trees for explainable k-means clustering," *Pattern Recognition*, vol. 137, p. 109239, 2023.

[6] D. Bertsimas, A. Orfanoudaki, and H. Wiberg, "Interpretable clustering: an optimization approach," *Machine Learning*, vol. 110, no. 1, pp. 89–138, 2021.

[7] P. Chasani and A. Likas, "Unsupervised decision trees for axis unimodal clustering," *Information*, vol. 15, no. 11, p. 704, 2024.

[8] 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.

[9] A. Spagnol, K. Sokol, P. Barbiero, M. Langheinrich, and M. Gjoreski, "Counterfactual explanations for clustering models," *arXiv preprint arXiv:2409.12632*, 2024.

[10] A. Karra, G. Vardakas, E. Pitoura, and A. Likas, "Generating counterfactual explanations for clustering models based on their equivalence to classification models," in *Artificial Intelligence Applications and Innovations*. Cham: Springer Nature Switzerland, 2025, pp. 85–100.

[11] 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.

[12] A. Van Looveren and J. Klaise, "Interpretable counterfactual explanations guided by prototypes," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2021, pp. 650–665.