Counterfactual explanations with the k-Nearest Neighborhood classifier and uncertain data

Emilio Carrizosa[a], Renato De Leone[b], Marica Magagnini[b,*]

[a] *Instituto de Matemáticas de la Universidad de Sevilla, Seville, 41012, Spain*
[b] *School of Science and Technology, Università di Camerino, Camerino, 62032, Italy*

Abstract

Counterfactual Analysis is a powerful tool in Explainable Machine Learning. Given a classifier and a record, one seeks the smallest perturbation necessary to have the perturbed record, called the counterfactual explanation, classified in the desired class. Feature uncertainty in data reflects the inherent variability and noise present in real-world scenarios, and therefore, there is a need to design Counterfactual Analysis methods that take into account uncertainty and provide robust counterfactual explanations. In this paper, we address the problem of finding counterfactual solutions for tabular data under uncertainty, where uncertainty is modeled assuming each record has a (convex) set around. The model is expressed as an optimization problem, which is solved with a Variable Neighborhood Search heuristic.

Keywords: Machine Learning, Counterfactual explanations, Decision making under uncertainty, k-NN, GaussVNS

1. Introduction

Uncertainty in data is a significant issue that may affect the quality of decisions and therefore addressing it properly is critical in data-driven decision making (Kläs and Vollmer, 2018; Alizadehsani et al., 2024; Fakour et al., 2024).

In classification problems under uncertainty in the input, one seeks robust classifiers not sensitive to small perturbations, where uncertainty is modeled as probability distributions or intervals around nominal values (Qin et al., 2009; Agrawal and Ram, 2015; Angiulli and Fassetti, 2013; Bertsimas et al., 2018; De Leone et al., 2023; Faccini et al., 2022).

*Corresponding author

Email addresses: ecarrizosa@us.es (Emilio Carrizosa), renato.deleone@unicam.it (Renato De Leone), marica.magagnini@unicam.it (Marica Magagnini)

Counterfactual Analysis (Wachter et al., 2018) is a powerful tool in Explainable Machine Learning, providing, for a classifier and a record, the so-called counterfactual explanation, which is the solution with minimal perturbation in the features that is classified in the desired class by the classifier. See (Carrizosa et al., 2024b) for a recent review on Counterfactual Analysis from a Mathematical Optimization perspective. Counterfactual explanations provide valuable information that allows the user to take action on his own features and thus change the prediction in a decision-making scenario. The information provided is user-friendly because it resembles contrastive human reasoning (Lipton, 1990) and is based on principles of similarity and proximity (Mothilal et al., 2020). In the literature, there are numerous methods for constructing counterfactual explanations, each focusing on a specific property of the classifier used or the explanations generated (Guidotti, 2022; Smyth and Keane, 2022; Carrizosa et al., 2024a; Verma et al., 2024).

Several open challenges have emerged from the instability of counterfactual explanations under small changes. Counterfactuals can be highly sensitive to perturbations or uncertainties, thus undermining their reliability in real-world applications (Slack et al., 2021; Virgolin and Fracaros, 2023). The definition of robustness can be model-dependent, with counterfactual explanations asked to fit different architectures exhibiting varying behaviors (Forel et al., 2024; Black et al., 2022; Dutta et al., 2022; Bui et al., 2022). Beyond instantaneous robustness, temporal stability has also been explored, looking consistency of the explanations over time (Ferrario and Loi, 2022). Robustness to input perturbations has been studied in (Artelt et al., 2021), while an analysis of the counterfactual explanation region of validity is provided in (Maragno et al., 2024).

In this paper, we present a new model for Counterfactual Analysis under uncertainty in the data, which are modeled as convex sets (Carrizosa et al., 2018). The classifier considered is the k-Nearest Neighborhood (k-NN), and therefore our model can be seen as the robust counterpart of the models recently introduced in (Magagnini et al., 2025; Contardo et al., 2024). The problem is written as a nonlinear mixed integer optimization problem, heuristically solved with a Gaussian Variable Neighborhood Search (Carrizosa et al., 2012).

We briefly describe our notation now. All vectors are column vectors where subscripts indicate components of a vector, while superscripts are used to identify different vectors. The set of real numbers will be denoted by $\mathbb{R}$, while the space of the J–dimensional vectors with real components will be indicated by $\mathbb{R}^J$. The symbol $\|x\|$ indicates the norm of the vector x. In particular, the Euclidean norm is denoted by $\|x\|_2 = \sqrt{x^\top x}$. Let $A \in \mathbb{R}^{J\times J}$ be a positive definite matrix, we define the ellipsoidal norm $\|x\|_{2,A} = \sqrt{x^\top A^\top A x} = \|Ax\|$. Similarly, $\|x\|_{\infty,A} \coloneqq \max_{j=1,\dots,J} |(Ax)_j|$. In the following, $\{(x^n, y_n)\}_{n=1,\dots,N}$ is a labeled dataset with $x^n \in \mathbb{R}^J$ features vector and $y_n \in \{0, 1\}$ label. We assume that categorical variables

have already been pre-processed with an appropriate encoding procedure, which transforms them into binary vectors.

The remainder of the paper is structured as follows. In Section 2 we introduce the k-NN classifier under uncertainty. Section 3 states the counterfactual explanation problem in case of uncertain data, while Section 4 analyses the very specific optimization problem for counterfactual explanations with the k-NN classifier and uncertain data. A constructive starting solution, and a tailored Variable Neighborhood Search are described in Section 5 as a solution procedure. An illustration of the experimental results is discussed in Section 6. The main contributions and some lines of research are resumed in the Conclusions, Section 7.

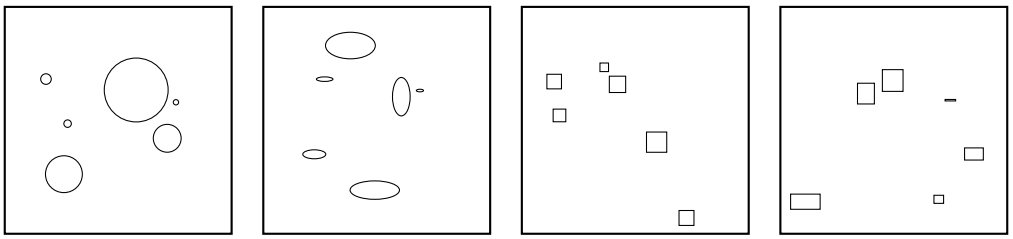
2. k-NN classifier with uncertain data

In this Section, we briefly describe how to adapt the traditional k-NN to the case in which instances are affected by uncertainty in their features.

To model uncertainty, we assign to each instance (x^n, y_n) in the training sample an uncertainty set U^n. We assume that each U^n is a convex set centered at x^n, more precisely each U^n is a ball (centered at x_n) of norm of the form $\|\cdot\|_{\ell,A^n}$ for $\ell \in \{2, \infty\}$ and a positive definite matrix $A^n \in \mathbb{R}^{J\times J}$:

$$U^n = \left\{x \in \mathbb{R}^J : x = x^n + u^n \text{ for some } u^n, \|u^n\|_{\ell,A^n} \leq 1\right\}.$$

This way, the original training sample $\{(x^1, y_1), \ldots, (x^n, y_n)\}$ is replaced by $\{(U^1, y_1), \ldots, (U^n, y_n)\}$. Since in the feature space each axis represents a feature, all instances must have the same axis orientation in their representation with uncertainty as convex sets. Then, we assume that all the symmetric positive definite matrices $A^{n\top}A^n$ share the same orthogonalization, meaning that their orthogonal vectors define the same axial orientation for all the convex sets. In particular, when $\ell = \infty$, the matrix A^n is restricted to be diagonal. Figure 1 shows some examples of the dataset with the embedded uncertainty.



(a) Balls ($\ell = 2$) (b) Ellispes ($\ell = 2$) (c) Squares ($\ell = \infty$) (d) Rectangles ($\ell = \infty$)

Figure 1: Possible dataset configurations according to the norm chosen for representing the uncertainty.

Given a new instance $\bar{x}$, with associated uncertainty set $\bar{U}$, the k-NN classification rule assigns to each point the majority class in its k closest neighbors. To implement it in our context with uncertainty, we need, therefore, a definition of distance for uncertainty sets, which, by construction, are convex compact sets.

We define the distance $d(U, \bar{U})$ between uncertain sets U and $\bar{U}$ as the minimum distance between points in the two sets,

$$d(U, \bar{U}) = \min_{v \in U, \bar{v} \in \bar{U}} \|\bar{v} - v\|_2, \tag{1}$$

see (Carrizosa et al., 2018), for more details and other possible choices and properties of distances between convex compact sets.

Given the training sample $\{(U^1, y_1), \ldots, (U^n, y_n)\}$ and the instance $\bar{U}$, let $r_k(\bar{U})$ be the k^{th}-smallest value in the set $\{d(U^1, \bar{U}), \ldots, d(U^N, \bar{U})\}$, and consider the set $\mathcal{N}_k(\bar{U})$ of instances at a distance not greater than $r_k(\bar{U})$,

$$\mathcal{N}_k(\bar{U}) = \{n : d(U^n, \bar{U}) \leq r_k(\bar{U})\}. \tag{2}$$

Note that $\mathcal{N}_k(\bar{U})$ has, by construction, at least k elements, but it may have more in case of ties.

Now, the k-NN classifier assigns a label 1 to $\bar{U}$ if at least a half of the instances in $\mathcal{N}_k(\bar{U})$ belongs to class 1, i.e., if

$$|\{n \in \mathcal{N}_k(\bar{U}) : y_n = 1\}| \geq \frac{1}{2}|\mathcal{N}_k(\bar{U})|. \tag{3}$$

3. Counterfactual explanations for uncertain data

This Section is devoted to generalize the problem of building counterfactual explanations for uncertain data.

Given a classification rule $f : \mathbb{R}^J \to \{0, 1\}$, the label associated to a new instance $\bar{x} \in \mathbb{R}^J$ is $\bar{y} = f(\bar{x})$. Building a counterfactual explanation for the couple $(\bar{x}, \bar{y})$ amounts to finding $x^c \in \mathbb{R}^J$ such that $f(x^c) \neq \bar{y}$ and the distance $d(x^c, \bar{x})$ between the input instance and its counterfactual is minimal (Wachter et al., 2018), where d is the distance in $\mathbb{R}^J$.

Consider the uncertainty set of $\bar{x}$, i.e.

$$\bar{U} = \{x \in \mathbb{R}^J \ : \ x = \bar{x} + \bar{u}, \ \|\bar{u}\|_{\ell, \bar{A}} \leq 1\},$$

its label is provided by a classification rule $F : \mathbb{R}^J \to \{0, 1\}$ such that $F(x) = F(\bar{x}) = \bar{y}$ for all $x \in \bar{U}$. Building a counterfactual explanation for $(\bar{U}, \bar{y})$ means determining

$$U^c = \{x \in \mathbb{R}^J \ : \ x = x^c + u^c, \ \|u^c\|_{\ell, \bar{A}} \leq 1\}$$

such that $F(x) = F(x^c) \neq \bar{y}$ for all $x \in U^c$ and the distance $d(U^c, \bar{U})$ is minimal, where now d is the distance defined in (1). Therefore, the counterfactual (U^c, y^c) is the optimal solution of the following optimization Problem

$$\begin{array}{ll} \min\limits_{U^c} & d(U,^c \bar{U}) \\ \text{s.t.} & F(U^c) = y^c \end{array} \tag{4}$$

where $y^c \neq y$.

4. k-NN optimal counterfactuals with uncertain data

This Section specifically describes Problem (4) in the case of the k-Nearest Neighborhood classifier.

Consider an input instance with uncertain features $(\bar{U}, \bar{y})$, where $\bar{y} = 0$ is the label given by the k-NN rule. The counterfactual explanation (U^c, y^c) has label $y^c = 1$ and it is the solution of Problem (4) when the classification function F is the k-NN rule as in (3):

$$\begin{array}{ll} \min\limits_{U^c} & d(U,^c \bar{U}) \\ \text{s.t.} & |\{n \in \mathcal{N}_k(U^c) : y_n = 1\}| \geq \dfrac{1}{2}|\mathcal{N}_k(U^c)|. \end{array} \tag{5}$$

For the rest of the paper, the labels of the input instance $\bar{U}$ and any associated counterfactual will be the same as the ones outlined here.

An extended version of Problem (5) in the classical no uncertainty case is described in (Magagnini et al., 2025). Here, we analyze the adaptation of the proposed formulation when we deal with data with uncertainty, and d is the minimal distance between two convex sets.

$$\min_{\substack{U^c, r_k \\ y_n^{\leq}, y_n^{<}}} \quad d(U^c, \bar{U}) \tag{6a}$$

$$\text{s.t.} \quad \sum_{n=1}^{N} y_n{}^{\leq} \cdot y_n \geq \frac{1}{2} \sum_{n=1}^{N} y_n{}^{\leq} \tag{6b}$$

$$d(U^c, U^n) - M(1 - y_n{}^{\leq}) \leq r_k \qquad n = 1, \ldots, N \tag{6c}$$

$$d(U^c, U^n) + M y_n{}^{\leq} \geq r_k + \epsilon \qquad n = 1, \ldots, N \tag{6d}$$

$$d(U^c, U^n) - M(1 - y_n{}^{<}) \leq r_k - \epsilon \qquad n = 1, \ldots, N \tag{6e}$$

$$d(U^c, U^n) + M y_n{}^{<} \geq r_k \qquad n = 1, \ldots, N \tag{6f}$$

$$\sum_{n=1}^{N} y_n{}^{\leq} \geq k \tag{6g}$$

$$\sum_{n=1}^{N} y_n{}^{<} \leq k - 1 \tag{6h}$$

$$y_n{}^{<}, y_n{}^{\leq} \in \{0, 1\} \qquad n = 1, \ldots, N \tag{6i}$$

$$r_k \in [0, M] \tag{6j}$$

Variable $r_k \coloneqq r_k(U^c)$ is the k^{th}-smallest distance in the set $\{d(U^c, U^1), \ldots, d(U^c, U^N)\}$, while $y_n^{\leq}$, $y_n^{<}$ are binary variables that identify whether an element U^n is in the k-neighborhood of the counterfactual $\mathcal{N}_k(U^c)$. Constraints (6c), (6d), (6e) and (6f) contribute to the localization of the counterfactual. Parameter $M \in \mathbb{R}$ is utilized for the linearization of the constraints, while ϵ is a sufficiently small quantity. The classification rule is expressed by constraints (6b), (6g) and (6h).

As we use the minimal Euclidean distance between two convex sets (1), Problem (6) becomes a bi-level minimization problem. Thus, we deepen the study of constraints (6c)-(6f). In the following we analyze the couple of constraints (6c)-(6d), the behavior of (6e)-(6f) being similar.

Despite $d(U^c, U^n)$ in (6c) is the minimal Euclidean distance, the constraint is equivalent to

$$\|x^c + u^c - (x^n + u^n)\|_2 - M(1 - y_n{}^{\leq}) \leq r_k \tag{7}$$

with $\|u^c\|_{\ell, \bar{A}} \leq 1$, $\|u^n\|_{\ell, A^n} \leq 1$. The counterfactual U^c and the instance U^n are represented by their centers x^c (a variable) and x^n (known) and by the uncertainty variables u^c and u^n, respectively. Removing the second-level minimization problem does not affect the requirement of minimum distance between two convex sets. When (7) holds, constraint (6c) is satisfied, too.

On the other hand, removing the second-level minimization problem in constraint (6d) is not straightforward. The direction of the inequality in the constraint prevents us from making the same considerations as in the previous case

with a minimization problem at the second level. Now, we transform the minimization problem into an equivalent maximization problem. The maximization problem at the second level can be eliminated without affecting the feasibility of the original constraint (6d).

We can observe that the following minimization Problems (8) are equivalent

$$\begin{aligned} \min_{u^c,u^n} \quad & \|x^c + u^c - (x^n + u^n)\|_2 \\ \text{s.t.} \quad & \|u^n\|_{\ell,A^n} \leq 1, \\ & \|u^c\|_{\ell,\bar{A}} \leq 1. \end{aligned} \tag{8}$$

$$\begin{aligned} \min_{u^c,u^n} \quad & \max_{\|z^n\|_2^2 \leq 1} z^{n\top}[x^c + u^c - (x^n + u^n)] \\ \text{s.t.} \quad & \|u^n\|_{\ell,A^n} \leq 1. \\ & \|u^c\|_{\ell,\bar{A}} \leq 1. \end{aligned}$$

where $z^n \in \mathbb{R}^J$. Hence, the minimization in constraint (6d) can be converted in the following

$$\begin{aligned} \max_{z^n} \quad & \left[z^{n\top}(x^c - x^n) - \max_{\|u^c\|_{\ell,\bar{A}} \leq 1} (-z^n)^\top u^c \; - \max_{\|u^n\|_{\ell,A^n} \leq 1} z^{n\top}(u^n) \right] \\ \text{s.t.} \quad & \|z^n\|_2^2 \leq 1, \end{aligned} \tag{9}$$

whose objective function has different forms according to the ℓ-norm. If $\ell = 2$, Problem (9) becomes

$$\max_{z^n} \quad z^{n\top}(x^c - x^n) - \|(\bar{A})^{-1} z^n\|_2 - \|(A^n)^{-1} z^n\|_2 \tag{10a}$$

$$\text{s.t.} \quad \|z^n\|_2^2 \leq 1, \tag{10b}$$

while, if $\ell = \infty$, it is

$$\max_{z^n} \quad z^{n\top}(x^c - x^n) - z^{n\top}(\bar{A})^{-1}\xi^n \; - z^{n\top}(A^n)^{-1}\xi^n \tag{11a}$$

$$\text{s.t.} \quad \|z^n\|_2^2 \leq 1, \tag{11b}$$

$$\xi^n = 2\gamma^n - 1, \tag{11c}$$

$$-M(1 - \gamma^n) \leq z^n \leq M\gamma^n - \epsilon, \tag{11d}$$

$$\gamma^n \in \{0, 1\}, \tag{11e}$$

where $\xi^n \in \{-1, 1\}^J$ and $\xi_j^n = 1$ if $z_j^n \geq 0$, $\xi_j^n = -1$ otherwise. Depending on the norm utilized, we can replace $d(U^c, U^n)$ in (6d) with the objective function (10a) ($\ell = 2$) and (11a) ($\ell = \infty$) and add to the upper level Problem (6), in both cases,

the constraints $\|z^n\|_2^2 \leq 1$, and constraints (11c)-(11e) when $\ell = \infty$. For $\ell = 2$, when

$$z^{n\top}(x^c - x^n) - \|(\bar{A})^{-1}z^n\|_2 - \|(A^n)^{-1}z^n\|_2 + My_n^{\leq} \geq r_k + \epsilon \tag{12}$$

holds, also constraint (6d) is satisfied. Similarly, for $\ell = \infty$ the constraint becomes

$$z^{n\top}(x^c - x^n) - z^{n\top}(\bar{A})^{-1}\xi^n \ - z^{n\top}(A^n)^{-1}\xi^n + My_n^{\leq} \geq r_k + \epsilon. \tag{13}$$

With the reformulation of the constraints as in (7) and (12)-(13), Problem (6) becomes a single-level optimization problem. However, finding an optimal solution for this problem is still challenging. In case of $\ell = 2$, in fact, (12) are Second Order Cone (SOC) constraints with a quadratic non-convex term, while for $\ell = \infty$, (13) are quadratic non-convex and their structure involves the addition of binary variables γ^n for $n = 1, \ldots, N$ in (11), which increases the complexity of the problem as the number of instances grows. The computational cost of transforming the bi-level problem into a single-level problem is excessively large, even disregarding the size of the dataset.

5. Heuristic solutions

Solving Problem (6) to optimality is too demanding even for datasets of small size. Therefore, we propose a heuristic method, which first constructs a feasible solution exploiting the structure of the problem, and then such solution is improved with a Variable Neighborhood Search.

5.1. Building a starting solution

We assume we can provide a feasible solution to Problem (6) looking for counterfactuals among the elements of the dataset, i.e. *endogenous* counterfactuals (Smyth and Keane, 2022). Actually, every U^n, for which the k-NN rule assigns label 1, is a possible counterfactual explanation for $\bar{U}$. The optimal choice is the one that minimizes the distance from the input $\bar{U}$:

$$d(U^e, \bar{U}) \leq d(U^n, \bar{U}) \quad n = 1, \ldots, N, \tag{14}$$

such that $y^e = F_{k\text{-NN}}(U^e) = 1$, $e \in \{1, \ldots, N\}$, where $F_{k\text{-NN}}$ is the k-NN rule that assigns a label according to the majority of the labels of the k nearest neighbors. Hence, computing the endogenous counterfactual means:

(a) finding the k nearest neighbors of U^n, among all $n = 1, \ldots, N$;
(b) checking if $F_{k\text{-NN}}(U^n) = 1$, for some $n = 1, \ldots, N$;
(c) selecting among all the U^n that satisfy (b), the one that minimizes the distance from $\bar{U}$ (14).

Point (a) could be computationally expensive as the size of the dataset increases because the construction of all the neighborhoods requires computing the distance between each pair of instances in the dataset. In (Contardo et al., 2024), the authors propose efficient strategies to reduce the computational cost of constructing a neighborhood.

The endogenous counterfactual represents the best feasible solution within the dataset for Problem (6), although it typically differs from the *exogenous* optimal solution. In the following Section, we propose a heuristic strategy that, starting from the endogenous counterfactual, constructs new feasible counterfactuals at shorter distances from the input compared to the initial endogenous one.

5.2. GaussVNS improvements

The heuristic strategy we propose follows the Gaussian Variable Neighborhood Search (GaussVNS) (Carrizosa et al., 2012), an extended methodology of the traditional Variable Neighborhood Search (VNS) (Mladenović and Hansen, 1997). Using GaussVNS, an incumbent counterfactual is perturbed by adding some noise generated from a Gaussian distribution. The perturbation of the incumbent aims to select a new neighborhood employed to build a better counterfactual. This counterfactual becomes the incumbent in the next iteration if it reduces the distance from the input compared to the current one.

To determine better counterfactuals, the algorithm systematically extends the research space by increasing the variance of the perturbation distribution till a maximum number G of choices, i.e. $\sigma = (\sigma_1, \ldots, \sigma_G)$, and it ends after a given time limit.

Let $(H, \mathcal{N}_k, \delta)$ be respectively the incumbent counterfactual $H = \{x \in R^J : x = x^H + u^H,\ \|u^H\|_{\ell,\bar{A}} \leq 1\}$, the set of the k nearest neighbors of H and the distance of H from $\bar{U}$, i.e. $\delta = d(H, \bar{U})$. Algorithm 1 is designed to construct better counterfactuals than the starting one $(H^s, \mathcal{N}_k^s, \delta^s)$, which is passed to the algorithm as the first incumbent. We suggest to use the endogenous counterfactual (Section 5.1) as initial incumbent of the heuristic, i.e. $(H^s, \mathcal{N}_k^s, \delta^s) = (U^e, \mathcal{N}_k^e, \delta^e)$, where $\mathcal{N}_k^e$ are the k nearest neighbors of the endogenous counterfactual and $\delta^e = d(U^e, \bar{U})$.

Let us delve into the details of an iteration of Algorithm 1. The heuristic first perturbs H according to the features type of the instance, see Algorithm 2. If the feature is categorical, the perturbation is a $\{0, 1\}$ coin flip. We refer in Algorithm 2 to the simple binary case, for multiple categories the strategy can be generalized. In the numerical case, τ_j is a random point from a Gaussian distribution $N(0, 1)$ with 0 mean and 1 variance, and the perturbation $\sigma_g \tau$ is applied as a translation of the center of the incumbent H.

Once the perturbed incumbent H^* is built, the heuristics computes the set of its k nearest neighbors $\mathcal{N}_k^*$. If the new set $\mathcal{N}_k^*$ is equal to the incumbent $\mathcal{N}_k$, then

Algorithm 1: Heuristic

Data: k, $\bar{U}$, H^s, $\mathcal{N}_k^s$, δ^s, G, $TimeLimit$
Result: H
$g = 1$; # Variance index #
$\lambda = 1$; # Research region magnitude #
$H = H^s$; # Incumbent counterfactual #
$\mathcal{N}_k = \mathcal{N}_k^s$; # k-Neighborhood of H #
$\delta = \delta^s$; # Distance $H - \bar{U}$ #
while $g \leq G$ ***and*** $t < TimeLimit$ **do**
  if $\lambda = 1$ **then**
    $H^* \leftarrow$ **Perturb** H;
    Compute $\mathcal{N}_k^*$;
  end
  if $\mathcal{N}_k^* = \mathcal{N}_k$ **then**
    Update g;
    $\lambda = 1$;
  else if $\mathcal{N}_k^*$ *not feasible* **then**
    Update g;
  else
    Compute $\mathcal{M}_k^*$;
    $\rho = \lambda \cdot d(H^*, U^*)$; # U^* is H^* k^{th}-NN #
    $H^{COP}, \delta^{COP} \leftarrow$ **Solve** $COP(H^*, \bar{U}, \mathcal{N}_k^*, \mathcal{M}_k^*, \rho)$;
    Compute $\mathcal{N}_k^{COP}$;
    if $\delta^{COP} < \delta$ ***and*** $\mathcal{N}_k^{COP}$ *feasible* **then**
      $H = H^{COP}$; # New closer incumbent #
      $\mathcal{N}_k = \mathcal{N}_k^{COP}$;
      $\delta = \delta^{COP}$;
      $g = 1$; # Reset #
      $\lambda = \lambda + 1$; # Enlarge research region #
    else
      Update g;
      $\lambda = 1$;
    end
  end
end

Algorithm 2: Perturb H

Data: H, σ_g
Result: H^*
for $j = 1, \ldots, J$ **do**
  if H_j *categorical* **then**
    $\tau_j \leftarrow$ **Random** $\{0, 1\}$; `# Coin flip #`
    $x_j^{H^*} = (x_j^H + \tau_j)\%2$;
  else
    $\tau_j \leftarrow$ **Random** $N(0, 1)$; `# Gaussian distribution #`
    $x_j^{H^*} = x_j^H + \sigma_g \tau_j$;
  end
end
$H^* = \{x \in \mathbb{R}^J \ : \ x = x^{H^*} + u^{H^*}, \ \|u^{H^*}\|_{\ell,\bar{A}} \leq 1\}$;

Algorithm 3: Update g

Data: g, G
Result: g
if $g < G$ **then**
  $g = g + 1$; `# Next variance value #`
else
  $g = 1$;
end

the current iteration terminates and in the next one H is perturbed using a larger variance (σ_{g+1}). Moreover, when $\mathcal{N}_k^* = \mathcal{N}_k$, g is updated according to Algorithm 3. Here, index g is incremented or reset to 1 if all the variance increments have already been explored.

If the new $\mathcal{N}_k^*$ is different from the previous $\mathcal{N}_k$, there is still to check that it is *feasible* to construct a counterfactual explanation. A *feasible* neighborhood to construct a counterfactual must have the majority of the instances with label 1. Again, if $\mathcal{N}_k^*$ is not *feasible*, the iteration stops and a different H^* is built in the next one with an updated variance (Algorithm 3).

If $\mathcal{N}_k^*$ is *feasible*, the heuristic computes the set $\mathcal{M}_k^*$ and the radius ρ. First, let $s_k(H^*)$ be the k^{th}-smallest value in the set $\{d(H^*, U^n) \ : \ n \notin N_k^*,\ y_n = 0\}$, then $\mathcal{M}_k^*$ is the set of the first k instances closest to H^* with label $y_n = 0$ which are not already in $\mathcal{N}_k^*$,

$$\mathcal{M}_k^* = \{n \ : \ r_k(H^*) < d(H^*, U^n) \leq s_k(H^*);\ y_n = 0\}.$$

On the other hand, radius ρ identifies a portion of the space, the *research region*, where to look for a new counterfactual. Actually, ρ is distance between $\bar{U}$ and the k^{th} nearest neighbor of H^* (called U^*) multiplied by a coefficient λ. We call λ the *research region magnitude*.

Problem (15) is an optimization problem that minimizes the distance from $\bar{U}$, when the solution is closer to the instances in $\mathcal{N}_k^*$ rather than to the ones in $\mathcal{M}_k^*$ and it belongs to $B(H^*, \rho)$, the ball centered in H^* center with radius ρ.

$$\begin{aligned}
\underset{U}{\text{minimize}} \quad & d(U, \bar{U}) && \text{(15a)}\\
\text{subject to} \quad & d(U, U^n) \leq d(U, U^m) - \epsilon \quad n \in \mathcal{N}_k^*, m \in \mathcal{M}_k^* && \text{(15b)}\\
& d(U, H^*) \leq \rho && \text{(15c)}
\end{aligned}$$

We will also refer to Problem (15) as COP, which is the optimization problem embedded in Algorithm 1 to generate instances candidate to be counterfactual. Problem (15) is much easier to solve than Problem (6). We will discuss in the next Sections two specific versions of (15): when the uncertain instances are rectangles ($\ell = \infty$) and when no uncertainty is provided (points).

At this point of the heuristic iteration, Problem (15) is solved by an optimal solver (within a time-limit of few seconds to increase the number of iterations of the heuristic) providing a feasible solution which is a possible new counterfactual H^{COP} and its distance δ^{COP} from $\bar{U}$. Actually, despite the restriction on the research region caused by ρ, the k nearest neighborhood of the solution (H^{COP}) could not be $\mathcal{N}_k^*$. Thus, the neighborhood of H^{COP} is computed $\mathcal{N}_k^{COP}$ to check if the majority of the instances have label 1 (*feasibility*), which confirms if H^{COP} is a counterfactual. When this happens and the new counterfactual H^{COP} is

closer to $\bar{U}$ ($\delta^{COP} < \delta$), then it becomes the new incumbent counterfactual, i.e. $(H, \mathcal{N}_k, \delta) = (H^{COP}, \mathcal{N}_k^{COP}, \delta^{COP})$. Finally, the variance index g is reset to 1 and the *research region magnitude* λ is incremented, so that in the next iteration no new neighborhood is computed but the same Problem (15) with a larger research region is solved. The search keeps focusing on the same research region, which is iteratively expanded through updates of λ, until the optimization Problem COP can no longer identify improvements. If a new counterfactual is found, but it does not improve the distance ($\delta^{COP} \geq \delta$), a new iteration of the heuristic will be started.

The heuristic terminates with the last best counterfactual computed along the iterations. This counterfactual is closer to the input $\bar{U}$ than the initial counterfactual provided to start Algorithm 1.

5.2.1. Variance vector σ

The variance vector σ is a parameter of Algorithm 1 that needs to be calculated before starting the heuristic. We suggest here a possible way to obtain it.

Let X be a random vector in $\mathbb{R}^J$, normally distributed with mean μ and covariance matrix $\sigma_1^2 I$, where I is the identity matrix. In the heuristic (Algorithm 1) X represents the incumbent perturbed counterfactual H^*, and μ stands for the actual incumbent H. Given a probability $\beta \in (0, 1)$, let us calculate $c(\beta) > 0$ satisfying

$$P(\|X - \mu\|_2 \leq c(\beta)) = \beta.$$

Considering the heuristic perspective, this means that the perturbed incumbent counterfactual has probability β of being in the ball centered on the actual incumbent and with radius that depends on β.

Observing that

$$\frac{1}{\sigma_1^2}\|X - \mu\|_2^2 = \sum_{j=1}^{J} \frac{1}{\sigma_1^2}(X_j - \mu_j)^2,$$

each term $\frac{1}{\sigma_1^2}\|X_j - \mu_j\|_2$ is an independent, normally distributed stochastic variable and follows a chi-square distribution with 1 degree of freedom (χ_1^2). Thus,

$$\frac{1}{\sigma_1^2}\|X - \mu\|_2^2 \sim \chi_J^2, \tag{16}$$

where χ_J^2 has J degrees of freedom. Hence, combining (16) with the following observation

$$\beta = P(\|X - \mu\|_2 \leq c(\beta)) \quad = \quad P\left(\frac{1}{\sigma_1^2}\|X - \mu\|^2 \leq \frac{c^2(\beta)}{\sigma_1^2}\right),$$

we can estimate

$$c(\beta) = \sigma_1\sqrt{\chi_{J,\beta}^2},$$

where $\chi^2_{J,\beta}$ is the β-quantile of a χ^2 distribution with J degrees of freedom.

Given $\mu = x^H$, the center of the incumbent solution in Algorithm 1, and c^* the distance from its nearest neighbor, we seek σ_1 such that the probability of generating a value of a normal distribution with mean μ and covariance $\sigma_1^2 I$ is at a distance from μ smaller than c^* with probability β, this is

$$\sigma_1 = \frac{c^*}{\sqrt{\chi^2_{J,\beta}}}.$$

Such σ_1 defines the first variance of the GaussVNS and $\sigma_g := 2^{g-1}\sigma_1$ for $g \in \{1, \ldots, G\}$.

Since our aim is to ensure the locality of the search of GaussVNS, high values of probability β are suggested, which means generating H^* close enough to the incumbent H. Moreover, by doubling the value of σ_1 at each step, we immediately allow the enlargement of the research space.

5.2.2. Rectangles: case $\ell = \infty$.

Here, we focus our attention on Problem (15) in the specific case where the convex set of uncertainty is a rectangle ($\ell = \infty$). Observe that the minimal distance between two rectangles can be calculated as the distance between a point and a rectangle. Consider, in fact, two rectangles U^n and U^m

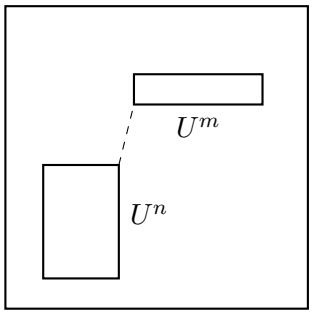
$$d(U^n, U^m) = \min_{\substack{w \in U^n \\ w' \in U^m}} \|w' - w\|_2 = \min_{\tilde{w} \in \tilde{U}^m} \|\tilde{w} - x^n\|_2$$

where $\tilde{U}^m = \{x \in \mathbb{R}^J \ : \ x = x^m + \tilde{u}^m, \ \|\tilde{u}^m_j\|_{\infty, \tilde{A}^m} \leq 1\}$ and $\tilde{A}^m$ is a diagonal positive definite matrix defined by the diagonal entries of A^n and A^m as follows:

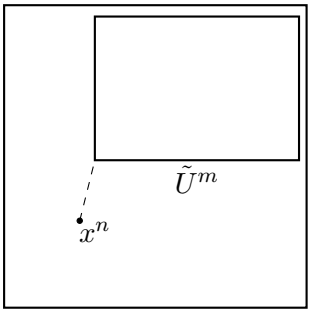
$$\tilde{A}^m_{jj} = \frac{A^n_{jj} A^m_{jj}}{A^m_{jj} + A^n_{jj}}. \tag{17}$$

Rectangle $\tilde{U}^m$ has the same center as U^m but a larger uncertainty which is the sum of U^n and U^m uncertainties (see Figure 2).

The advantage of measuring the distance between two rectangles as point-rectangle lies in reducing the number of variables that describe the uncertainty, which is a computational improvement for Problem (15). From now on, when we calculate the distance between a counterfactual U^c (as well as the incumbent H or the perturbed H^* in Algorithm 1) and any instance of the dataset U^m, we will always use the point-rectangle approach. The center of the counterfactual x^c will be the point, while the rectangle will be the uncertainty set centered at the instance of the dataset with the sum of both the uncertainties, and we will call it $\tilde{U}^m$. Since we assume in this work that the counterfactual has the



(a) $d(U^n, U^m)$

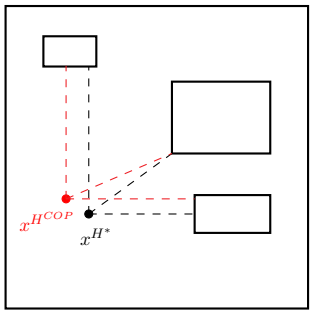


(b) $d(x^n, \tilde{U}^m)$

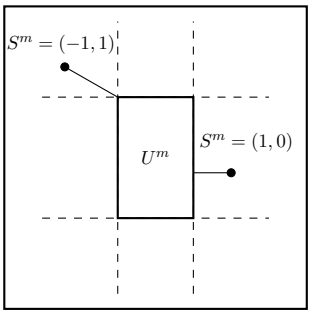
Figure 2: The distance between the two rectangles U^n and U^m in (a) is equal to the point-rectangle $(x^n, \tilde{U}^m)$ distance in (b), where the rectangle $\tilde{U}^m$ includes both the uncertainties of U^n and U^m.

same uncertainty as the input instance $\bar{U}$, the matrix $\tilde{A}^m$ is computed as in (17) considering the matrices A^m and $\bar{A}$.

A further computational improvement can be achieved by observing that the point-rectangle distance is always realized either between the point and a vertex or between the point and an edge of the rectangle. Therefore, in cases where this information can be precomputed, the complexity of the point-rectangle distance problem can be reduced. In Algorithm 1, the perturbed incumbent H^* is used to select a new *feasible* neighborhood $\mathcal{N}_k^*$ and the associated $\mathcal{M}_k^*$, which are passed to the optimization Problem (15). We can calculate a priori the distance between H^* and any instance in $\mathcal{M}_k^*$, we observe which vertex or edge is involved in the realization of this distance and we can constraint the solution H^{COP} to have the same relative positions to the instances in $\mathcal{M}_k^*$ as H^*, see Figure 3a. Consider



(a) H^* and H^{COP} sharing the same relative positions with the rectangles in the figure.



(b) Point-rectangle relative positions ($j = 2$) translated to S^m vector (18).

Figure 3: The minimal distance between a point and a rectangle always involves an edge or a vertex of the rectangle.

H^* and $m \in \mathcal{M}_k^*$, as in Algorithm 1, their distance is

$$d(H^*, U^m) = \min_{\tilde{w} \in \tilde{U}^m} \|\tilde{w} - x^{H^*}\|_2 = \min_{\|\tilde{u}^m\|_{\infty, \tilde{A}^m} \leq 1} \left[\sum_{j=1}^{J} (x_j^m + \tilde{u}_j^m - x_j^{H^*})^2 \right]^{\frac{1}{2}},$$

where the matrix $\tilde{A}^m$ associated to $\tilde{U}^m$ is a diagonal positive definite matrix defined by A^m and $\bar{A}$ as in (17). To simplify the notation, we call the diagonal entries $a_j^m \coloneqq \tilde{A}_{jj}^m$. Now, let define Δ_j as follow

$$\Delta_j \coloneqq \min_{\|\tilde{u}^m\|_{\infty, \tilde{A}^m} \leq 1} (x_j^m + \tilde{u}^m - x_j^{H^*})^2 = \begin{cases} (x_j^m - \frac{1}{a_j^m} - x_j^{H^*})^2 & \text{if} \quad x_j^{H^*} - x_j^m < -\frac{1}{a_j^m}, \\ 0 & \text{if} \quad |x_j^{H^*} - x_j^m| \leq \frac{1}{a_j^m}, \\ (x_j^m + \frac{1}{a_j^m} - x_j^{H^*})^2 & \text{if} \quad x_j^{H^*} - x_j^m > \frac{1}{a_j^m}, \end{cases}$$

with x^{H^*} and x^m the centers of H^* and $\tilde{U}^m$, respectively. Then,

$$d(H^*, U^m) = \left[\sum_{j=1}^{J} \Delta_j \right]^{\frac{1}{2}}$$

overtakes the minimization problem and directly calculates the distance. Moreover, from the definition of Δ_j we can extract a rule to record the relative positions between H^* and every instance U^m, $m \in \mathcal{M}_k^*$. Let define a vector $S^m \in \{-1, 0, 1\}^J$ such that

$$S_j^m = \begin{cases} -1 & \text{if} \quad x_j^{H^*} - x_j^m < -\frac{1}{a_j^m} \\ 0 & \text{if} \quad |x_j^{H^*} - x_j^m| \leq \frac{1}{a_j^m} \\ 1 & \text{if} \quad x_j^{H^*} - x_j^m > \frac{1}{a_j^m} \end{cases} \tag{18}$$

for $m \in \mathcal{M}_k^*$. A graphical intuition of the construction of the S^m vector is given in Figure 3b. The following reformulation of Problem (15) takes into account the information on S^m relative to the positions between H^* and the instances in $\mathcal{M}_k^*$:

$$\underset{x}{\text{minimize}} \quad d(x, \bar{U}) \tag{19a}$$

$$\text{subject to} \quad d(x, U^m) \leq \sum_{j=1}^{J} |S_j^m| (x_j^m + \frac{S_j^m}{a_j^m} - x_j)^2 - \epsilon \quad n \in \mathcal{N}_k^*,\ m \in \mathcal{M}_k^* \tag{19b}$$

$$d(x, H^*) \leq \rho \tag{19c}$$

$$x \in X_S \tag{19d}$$

where the second level minimization problem in (19b) and (19c) due to the minimal distance can be directly removed without loss of generality. Constraint (19d) imposes that

$$\begin{cases} x_j - x_j^m < -\frac{1}{a_j^m} & \text{if} \quad S_j^m = -1, \\ |x_j - x_j^m| \leq \frac{1}{a_j^m} & \text{if} \quad S_j^m = 0, \\ x_j - x_j^m > \frac{1}{a_j^m} & \text{if} \quad S_j^m = 1, \end{cases}$$

embedding the positional information inside the optimization problem. This modified version of Problem (15) is more time-efficient and can be employed in Algorithm 1 as the COP Problem. The numerical results of this implementation will be discussed in Section 6.

5.2.3. Point case

It is worth mentioning the particular case of a problem without uncertainty (points). Actually, Problem (15) can be formulated just considering the Euclidean distance and the original dataset:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \|x - \bar{x}\|_2^2 && \text{(20a)} \\ \text{subject to} \quad & \|x - x^n\|_2^2 \leq \|x - x^m\|_2^2 - \epsilon \quad n \in N_k^*,\ m \in \mathcal{M}_k^* && \text{(20b)} \\ & \|x - x^{H^*}\|_2^2 \leq \rho && \text{(20c)} \end{aligned}$$

Problem (20) is particularly easy to solve and thus suitable for combination with Algorithm 1. Therefore, in the case of no uncertainty, the heuristic allows the construction of good-quality counterfactuals that approximate the optimal one. The average time is shorter compared to the computation of the optimal solution. We will illustrate some comparisons in the following experiment section.

6. Computational Experiments

In this Section, we apply Algorithm 1 on a real dataset, showing how the heuristic works. The discussion will involve only the keynote experiments on the *Boston Housing* dataset (Harrison and Rubinfeld, 1978). A brief description of the dataset is given in Appendix A. Further analyses are available on GitHub at https://github.com/MagagniniMarica/Uncertain-Data-kNN-Counterfactuls.

All the numerical experiments have been performed on a PC, with an Intel R CoreTM i7-12650H CPU @ 2.30 GHz processor and 16 gigabytes RAM. The operating system is Windows 11, 64 bits. Algorithm 1 has been implemented in Python 3.12 while for the COP optimization problems, we use pyomo optimization modeling language (Hart et al., 2011; Bynum et al., 2021) and Gurobi 12.0 solver (Gurobi Optimization, LLC, 2024).

We designed 3 main experiments involving 10 distinct input instances (see Table A.3 in Appendix A), each already classified with label 0 by the k-NN rule. For each instance, we seek counterfactual explanations with label 1. We report the average results from 10 runs of Algorithm 1 for each instance. The variance vector σ for all these experiments is computed as proposed in Section 5.2.1 with $G = 4$ and $\beta = 0.99$. The COP Problem in Algorithm 1 is solved with a time limit of 5 seconds, which is usually sufficient to reach an optimal solution. However, when this is not the case, a good feasible solution is still acceptable, as the overall goal of the algorithm is a local search focused on improving the current incumbent quickly. The analysis that follows is a comparison of the distances between the input instances and the counterfactuals built with different methods.

The first experiment is a validation of the methodology. We tested the heuristic on the basic normalized *Boston housing* (BH) dataset with no uncertainty, comparing the results obtained using Algorithm 1 with the optimal solutions provided by (Magagnini et al., 2025) methodology, i.e., Problem (6) with Euclidean distance. For different values of k in the k-NN rule ($k = 3, 5, 7$), we computed the endogenous counterfactual (x^e), the heuristic one (x^H), and the actual optimal (x^O). To speed up the computations, in all the following experiments we computed the counterfactuals as described in Section 5.1 but choosing only among the instances already labeled as 1. The heuristic counterfactuals are provided by Algorithm 1 combined with (20) as COP Problem (parameter $\epsilon = 0.00001$) and a time-limit of 150 seconds per run. Moreover, counterfactual x^H results from the heuristic Algorithm 1 with the endogenous counterfactual x^e as initial incumbent. In Table 1 are reported the distances between the three kinds of counterfactuals and the input instances, i.e., endogenous δ^e, heuristic δ^H (average distance of 10 runs) and optimal δ^O.

$\bar{x}$	**k=3**			**k=5**			**k=7**		
	δ^e	δ^H	δ^O	δ^e	δ^H	δ^O	δ^e	δ^H	δ^O
1	0.451395	0.212454	0.201083	0.451395	0.184499	0.183829	0.451395	0.097131	0.068305
2	0.283759	0.016704	0.016659	0.321424	0.025165	0.025289	0.321424	0.036173	0.029402
3	0.285653	0.111586	0.095542	0.285653	0.080511	0.080737	0.285653	0.094930	0.093555
4	0.611435	0.302817	0.293881	0.611435	0.222817	0.222818	0.611435	0.131217	0.131216
5	0.307697	0.124082	0.113681	0.321076	0.128207	0.110140	0.321076	0.109558	0.067406
6	0.843765	0.534686	0.503927	0.843765	0.656749	0.577679	0.843765	0.714399	0.630409
7	0.742691	0.494566	0.393258	0.742691	0.441368	0.441667	0.783416	0.568918	0.486145
8	0.807388	0.625774	0.466810	0.807388	0.637607	0.548199	0.807388	0.650620	0.587915
9	0.927064	0.575255	0.462193	0.927064	0.541029	0.510952	0.927064	0.509621	0.509622
10	0.454322	0.454322	0.201629	0.454322	0.454322	0.237395	0.454322	0.454322	0.264120

Table 1: Distances comparison between each of the 10 instances $\bar{x}$ and the endogenous counterfactual (δ^e), the heuristic counterfactual (δ^H) provided by Algorithm 1 with COP Problem (20) and the optimal counterfactual (δ^O). Number of nearest neighbors $k = 3, 5, 7$.

We observe that the heuristic is almost always able to improve the initial

counterfactual, in this case the endogenous, and constructs counterfactuals whose distance from the input instance (δ^H) is close to that of the optimal counterfactual (δ^O). Input 10 is the only one for which the heuristic did not achieve any improvement. However, several times the heuristic x^H converges to the optimal solution x^O. Actually, when comparing the neighborhoods, x^O and x^H typically share the same neighbors.

To visualize the quality of the improvement of x^H, starting from x^e, we use x^O as the reference value. The closer the distance δ^H is to the distance δ^O, the better is the quality of the counterfactual x^H and the greater is the improvement obtained by the heuristic method. Let

$$\frac{\delta^e - \delta^H}{\delta^e - \delta^O} \cdot 100 \tag{21}$$

denote the percentage improvement in distance δ^H, i.e., the percentage by which x^H approaches x^O starting from x^e. Except for instance 10, each heuristic counter-

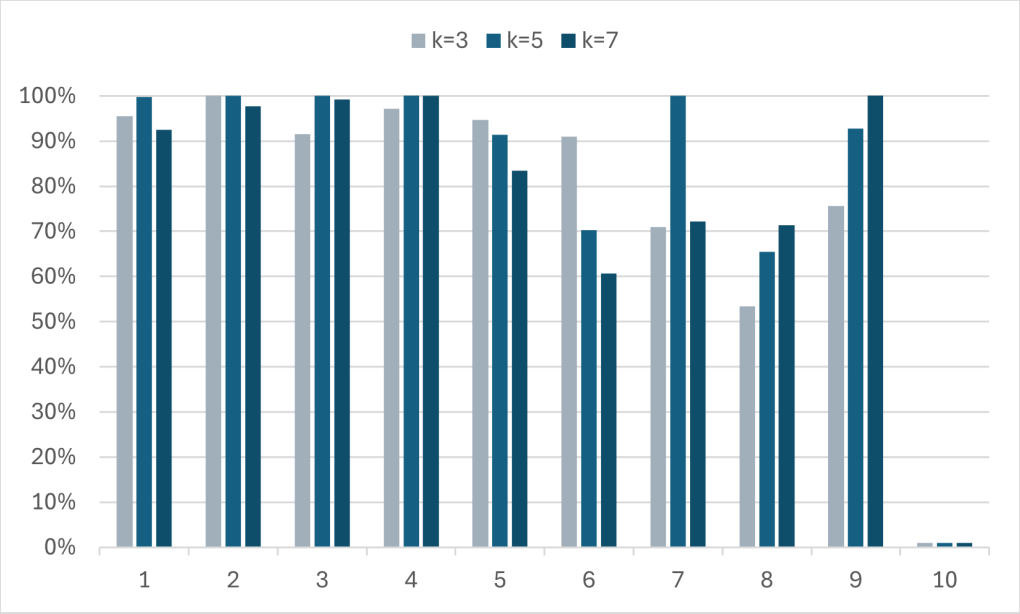


Figure 4: Heuristic average percentage improvement (21) per each instance $\bar{x}$ for three different values of $k = 3, 5, 7$.

factual improves the distance from the endogenous to the optimal counterfactual by at least 50%. In some cases, the heuristic counterfactual is exactly the optimal, in Figure 4 some improvements reach the 100%. Moreover, the quality of the improvements does not depend on the parameter k. The improvements that can be seen in Figure 4 are produced in less than a minute on average independently

from the choice of k, while, as stated in (Magagnini et al., 2025), the optimal counterfactual is constructed in twice the time as the number k of neighbors increases and depends on the hyperparameters of the solver. The heuristic provides good results in a shorter time that depends neither on the classifier parameter k nor on the hyperparameters of the solver.

The other two experiments introduce the uncertainty in the `Boston Housing` dataset. The instances of the dataset become the centers of the convex sets in the dataset with uncertainty $\mathcal{U}_{BH}$. For each numerical feature j, the uncertainty bounds are the 10% of the average over all values of that feature in the dataset:

$$\frac{1}{\theta_j} = \frac{1}{10}\left(\frac{1}{N}\sum_{n=1}^{N} x_j^n\right),$$

for all $n = 1, \ldots, N$, the uncertainty variables satisfy

$$-\frac{1}{\theta_j} \leq u_j^n \leq \frac{1}{\theta_j},$$

and all the matrices $A^n = diag(\theta_j, \ldots, \theta_J)$. The categorical instance "CHAS" has no uncertainty. Dataset $\mathcal{U}_{BH}$ consists of identical hyperrectangles of dimension $J-1$ since no uncertainty is associated with the categorical variable "CHAS". Similarly, the 10 inputs instances $\bar{x}$ in Table A.3 are the centers of the associated inputs with uncertainty $\bar{U}$ in which variables $(\bar{u})$ have the same uncertainty bounds as the instances in $\mathcal{U}_{BH}$, i.e. $-\frac{1}{\theta_j} \leq \bar{u}_j \leq \frac{1}{\theta_j}$ for all $j = 1, \ldots, J$ and $\bar{A} = A^n$. Note that, since the dataset is normalized, the features are all constrained to a certain interval. When we apply Algorithm 1, in particular during the perturbation phase (Algorithm 2), if the perturbed feature exceeds the boundaries, it is projected onto the feasible set.

$\bar{U}$	**k=3**		**k=5**		**k=7**	
	δ^e	δ^H	δ^e	δ^H	δ^e	δ^H
1	0.288357	0.127033	0.288357	0.155445	0.288357	0.187571
2	0.148401	0.079862	0.114627	0.065728	0.114627	0.078417
3	0.064473	0.039252	0.064473	0.020604	0.064473	0.018490
4	0.519705	0.263821	0.519705	0.168101	0.519705	0.194188
5	0.127234	0.051444	0.127234	0.053438	0.127234	0.060325
6	0.811572	0.609563	0.713839	0.662531	1.143533	0.987550
7	0.668415	0.485063	0.636282	0.577758	1.096228	0.890192
8	0.783076	0.640472	0.710147	0.654515	1.156458	0.896418
9	0.795444	0.532812	0.770648	0.593853	1.038946	0.885158
10	0.252789	0.154883	0.252789	0.178313	0.252789	0.189258

Table 2: For each input $\bar{U}$ and k neighbors, the two columns show the endogenous distance δ^e and the heuristic average distance δ^H from the input $\bar{U}$. Number of nearest neighbors involved $k = 3, 5, 7$.

For different numbers of nearest neighbors in k-NN rule ($k = 3, 5, 7$), we compute endogenous counterfactual U^e and we use it as first incumbent for Algorithm 1 combined with COP Problem (19), which provides a heuristic counterfactual H. For Algorithm 1, a run time limit was set to 150 seconds. Table 2 shows the distance results associated with the endogenous (δ^E) and the heuristic counterfactual (δ^H) as the average of 10 runs. Even though we don't have a reference point as in the previous case of no uncertainty, we can observe that the heuristic never fails to find new counterfactuals that are closer to the input than the endogenous one. Actually, the δ^H distances are always lower than the δ^e ones.

Finally, with the same experiment setting, we compare the improvements for longer run times. Figure 5 shows what happens to distances δ^H when the time limit is increased to 150, 300, and 600 seconds. When $k = 3, 5$, we observe a clear

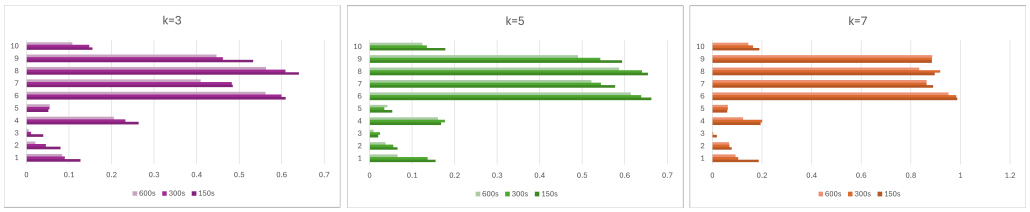


Figure 5: For each input $\bar{U}$ and k neighbors, the plot shows the average distance δ^H with 3 different time limit, i.e. 150, 300, and 600 seconds. Number of nearest neighbors involved $k = 3, 5, 7$.

pattern: distances are shorter the longer the run time limit is. For $k = 7$, this is not that obvious, actually a bigger k seems to put more strain on the heuristic, which therefore needs a longer time limit. The lack of improvements does not imply that the heuristic solutions are close to the true optimal solutions, but rather that by increasing the number of neighbors that classify the counterfactual, it becomes more difficult for the heuristic to select better neighborhoods.

7. Conclusions

The paper develops a model to construct robust counterfactual explanations under uncertainty in the data. As in any Counterfactual Analysis methodology, a classification rule is given, and for the analysis in this paper, the k-NN classifier is chosen, although our approach can be extended to other choices of the classifier. The uncertainty has been modeled as convex sets of different shapes, namely rectangles ($\ell = \infty$) and ellipses ($\ell = 2$). The problem has been modeled as a bi-level optimization problem, which resulted too hard to be solved even when converted into a classical single-level problem. Second, a heuristic is designed. A starting solution is built by finding the endogenous counterfactual, i.e., when the

search space is reduced to the instances in the training set. From such a starting solution, a Gaussian VNS is used.

Some experiments are performed to explore the performance of our model and algorithm by presenting some numerical results for the case of rectangular uncertainty ($\ell = \infty$). The $\ell = 2$ balls case presents similar complexities. The COP associated Problem can be reformulated, avoiding integer variables, and solved with continuous nonlinear solvers.

Two extensions of the model considered deserve further investigation. First, the shape of the uncertainty sets has been considered to be fixed, while it could be considered, at the expense of increasing the complexity of the optimization problem, as decision variables. Finally, the distance between the uncertainty set and its counterfactual is assumed to be given by the Euclidean distance. A combined distance combining not only the Euclidean, but also the ℓ_0 norm and asymmetric distances, as in Carrizosa et al. (2024b), would be an extension giving more realistic and interpretable decisions.

Acknowledgements

The work of one of the authors has been funded by the European Union - NextGenerationEU, Mission 4, Component 2, under the Italian Ministry of University and Research (MUR) National Innovation Ecosystem grant ECS00000041 - VITALITY - CUPJ13C22000430001. For the same author, this study was also carried out under the project INdAM – GNCS 2024 - CUP E53C23001670001. The manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

Appendix A. Boston Housing

The *Boston Housing* dataset consists of $N = 506$ instances and $J = 13$ features, 12 being numerical and 1 ("CHAS") categorical (binary). A description of the features can be found in (Harrison and Rubinfeld, 1978). In Section 6, the experiments use the normalized [0,1] version of the dataset. The target is a binary variable $\{0, 1\}$, i.e., it has value 1 if the price of the house is higher than the median value and 0 otherwise.

The following table reports 10 input instances of 0 label assigned by the k-NN rule on *Boston Housing*.

$\bar{x}$	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
1	0.001840	0.125	0.2716285	0.0	0.286008	0.468097	0.8547889	0.496731	0.173913	0.236641	0.276596	0.974305	0.424117
2	0.002399	0.0	0.236437	0.0	0.129630	0.391071	0.608651	0.450863	0.086957	0.087786	0.563830	1.0	0.399283
3	0.001607	0.25	0.171188	0.0	0.139918	0.417705	0.651905	0.554320	0.304348	0.185115	0.755319	0.995486	0.315121
4	0.015820	0.0	0.700880	1.0	1.0	0.492048	0.958805	0.056361	0.173913	0.412214	0.223404	0.808664	0.369481
5	0.002870	0.0	0.346041	0.0	0.327160	0.471738	0.901133	0.154989	0.130435	0.223282	0.617021	0.998487	0.275662
6	0.124781	0.0	0.646628	0.0	0.582305	0.257712	1.0	0.004056	1.0	0.914122	0.808511	1.0	0.911700
7	0.274109	0.0	0.646628	0.0	0.648148	0.209044	1.0	0.030700	1.0	0.914122	0.808511	1.0	0.732616
8	0.430994	0.0	0.646628	0.0	0.633745	0.362522	1.0	0.032736	1.0	0.914122	0.808511	1.0	0.796358
9	0.137585	0.0	0.646628	0.0	0.409465	0.436099	0.584964	0.078931	1.0	0.914122	0.808511	0.061350	0.385210
10	0.0031849	0.0	0.338343	0.0	0.411523	0.350450	0.720906	0.151770	0.217391	0.389313	0.702128	1.0	0.535596

Table A.3: 10 input instances feature values for the *Boston Housing* dataset.

References

Agrawal, R., Ram, B., 2015. A modified k-nearest neighbor algorithm to handle uncertain data, in: 2015 5th International Conference on IT Convergence and Security (ICITCS), pp. 1–4. doi:10.1109/ICITCS.2015.7292920.

Alizadehsani, R., Roshanzamir, M., Hussain, S., Khosravi, A., Koohestani, A., Zangooei, M.H., Abdar, M., Beykikhoshk, A., Shoeibi, A., Zare, A., Panahiazar, M., Nahavandi, S., Srinivasan, D., Atiya, A.F., Acharya, U.R., 2024. Handling of uncertainty in medical data using machine learning and probability theory techniques: a review of 30 years (1991–2020). Annals of Operations Research 339, 1077–1118. doi:10.1007/s10479-021-04006-2.

Angiulli, F., Fassetti, F., 2013. Nearest neighbor-based classification of uncertain data. ACM Transactions on Knowledge Discovery from Data 7. doi:10.1145/2435209.2435210.

Artelt, A., Vaquet, V., Velioglu, R., Hinder, F., Brinkrolf, J., Schilling, M., Hammer, B., 2021. Evaluating robustness of counterfactual explanations, in: 2021 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 01–09. doi:10.1109/SSCI50451.2021.9660058.

Bertsimas, D., Dunn, J., Pawlowski, C., Zhuo, Y., 2018. Robust classification. INFORMS Journal on Optimization 1. doi:10.1287/ijoo.2018.0001.

Black, E., Wang, Z., Fredrikson, M., 2022. Consistent counterfactuals for deep models, in: International Conference on Learning Representations. URL: https://doi.org/10.48550/arXiv.2110.03109.

Bui, N., Nguyen, D., Nguyen, V.A., 2022. Counterfactual plans under distributional ambiguity, in: International Conference on Learning Representations. URL: https://doi.org/10.48550/arXiv.2201.12487.

Bynum, M.L., Hackebeil, G.A., Hart, W.E., Laird, C.D., Nicholson, B.L., Siirola, J.D., Watson, J.P., Woodruff, D.L., 2021. Pyomo–optimization modeling in python. volume 67. Third ed., Springer Science & Business Media.

Carrizosa, E., Dražić, M., Dražić, Z., Mladenović, N., 2012. Gaussian variable neighborhood search for continuous optimization. Computers & Operations Research 39, 2206–2213. doi:https://doi.org/10.1016/j.cor.2011.11.003.

Carrizosa, E., Guerrero, V., Romero Morales, D., 2018. Visualizing data as objects by dc (difference of convex) optimization. Mathematical Programming 169, 119–140. doi:10.1007/s10107-017-1156-1.

Carrizosa, E., Ramírez-Ayerbe, J., Romero Morales, D., 2024a. Generating collective counterfactual explanations in score-based classification via mathematical optimization. Expert Systems with Applications 238, 121954. doi:https://doi.org/10.1016/j.eswa.2023.121954.

Carrizosa, E., Ramírez-Ayerbe, J., Romero Morales, D., 2024b. Mathematical optimization modelling for group counterfactual explanations. European Journal of Operational Research 319, 399–412. doi:https://doi.org/10.1016/j.ejor.2024.01.002.

Contardo, C., Fukasawa, R., Rousseau, L.M., Vidal, T., 2024. Optimal counterfactual explanations for k-nearest neighbors using mathematical optimization and constraint programming, in: Basu, A., Mahjoub, A.R., Salazar González, J.J. (Eds.), Combinatorial Optimization, Springer Nature Switzerland, Cham. pp. 318–331. doi:10.1007/978-3-031-60924-4_24.

De Leone, R., Maggioni, F., Spinelli, A., 2023. A robust twin parametric margin support vector machine for multiclass classification. doi:10.2139/ssrn.4793505.

Dutta, S., Long, J., Mishra, S., Tilli, C., Magazzeni, D., 2022. Robust counterfactual explanations for tree-based ensembles, in: Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S. (Eds.), Proceedings of the 39th International Conference on Machine Learning, PMLR, Baltimore, Maryland, USA. pp. 5742—-5756.

Faccini, D., Maggioni, F., Potra, F.A., 2022. Robust and distributionally robust optimization models for linear support vector machine. Computers & Operations Research 147, 105930. doi:https://doi.org/10.1016/j.cor.2022.105930.

Fakour, F., Mosleh, A., Ramezani, R., 2024. A structured review of literature on uncertainty in machine learning & deep learning. doi:10.48550/arXiv.2406.00332.

Ferrario, A., Loi, M., 2022. The robustness of counterfactual explanations over time. IEEE Access 10, 82736–82750. doi:10.1109/ACCESS.2022.3196917.

Forel, A., Parmentier, A., Vidal, T., 2024. Don't explain noise: Robust counterfactuals for randomized ensembles, in: Dilkina, B. (Ed.), Integration of Constraint Programming, Artificial Intelligence, and Operations Research, Springer Nature Switzerland, Cham. pp. 293–309. doi:10.1007/978-3-031-60597-0_19.

Guidotti, R., 2022. Counterfactual explanations and how to find them: literature review and benchmarking. Data Mining and Knowledge Discovery 38, 2770–2824. doi:10.1007/s10618-022-00831-6.

Gurobi Optimization, LLC, 2024. Gurobi Optimizer Reference Manual. URL: https://www.gurobi.com.

Harrison, D., Rubinfeld, D., 1978. Hedonic housing prices and the demand for clean air. Journal of Environmental Economics and Management 5, 81–102. doi:10.1016/0095-0696(78)90006-2.

Hart, W.E., Watson, J.P., Woodruff, D.L., 2011. Pyomo: modeling and solving mathematical programs in python. Mathematical Programming Computation 3, 219–260.

Kläs, M., Vollmer, A.M., 2018. Uncertainty in machine learning applications: A practice-driven classification of uncertainty, in: Gallina, B., Skavhaug, A., Schoitsch, E., Bitsch, F. (Eds.), Computer Safety, Reliability, and Security, Springer International Publishing, Cham. pp. 431–438.

Lipton, P., 1990. Contrastive explanation. Royal Institute of Philosophy Supplement 27, 247–266. doi:10.1017/s1358246100005130.

Magagnini, M., Carrizosa, E., De Leone, R., 2025. Nearest neighbors counterfactuals, in: Nicosia, G., Ojha, V., Giesselbach, S., Pardalos, M.P., Umeton, R. (Eds.), Machine Learning, Optimization, and Data Science, Springer Nature Switzerland, Cham. pp. 193–208. doi:10.1007/978-3-031-82481-4_14.

Maragno, D., Kurtz, J., Röber, T.E., Goedhart, R., Birbil, Ş.İ., den Hertog, D., 2024. Finding regions of counterfactual explanations via robust optimization. INFORMS Journal on Computing 36, 1316–1334. doi:10.1287/ijoc.2023.0153.

Mladenović, N., Hansen, P., 1997. Variable neighborhood search. Computers & Operations Research 24, 1097–1100. doi:https://doi.org/10.1016/S0305-0548(97)00031-2.

Mothilal, R.K., Sharma, A., Tan, C., 2020. Explaining machine learning classifiers through diverse counterfactual explanations, in: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, Association for Com-

puting Machinery, New York, NY, USA. p. 607–617. doi:10.1145/3351095.3372850.

Qin, B., Xia, Y., Prabhakar, S., Tu, Y., 2009. A rule-based classification algorithm for uncertain data, in: 2009 IEEE 25th International Conference on Data Engineering, pp. 1633–1640. doi:10.1109/ICDE.2009.164.

Slack, D., Hilgard, S., Lakkaraju, H., Singh, S., 2021. Counterfactual explanations can be manipulated, in: Proceedings of the 35th International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA. p. 62–75.

Smyth, B., Keane, M.T., 2022. A few good counterfactuals: Generating interpretable, plausible and diverse counterfactual explanations, in: Keane, M.T., Wiratunga, N. (Eds.), Case-Based Reasoning Research and Development, Springer International Publishing, Cham. pp. 18–32. doi:10.1007/978-3-031-14923-8_2.

Verma, S., Boonsanong, V., Hoang, M., Hines, K., Dickerson, J., Shah, C., 2024. Counterfactual explanations and algorithmic recourses for machine learning: A review. ACM Computing Surveys 56. doi:10.1145/3677119.

Virgolin, M., Fracaros, S., 2023. On the robustness of sparse counterfactual explanations to adverse perturbations. Artificial Intelligence 316, 103840. doi:https://doi.org/10.1016/j.artint.2022.103840.

Wachter, S., Mittelstadt, B., Russell, C., 2018. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. Harvard journal of law & technology 31, 841–887. doi:10.2139/ssrn.3063289.