Data Mining and Knowledge Discovery
https://doi.org/10.1007/s10618-022-00831-6

Counterfactual explanations and how to find them: literature review and benchmarking

Riccardo Guidotti[1]

Received: 1 April 2021 / Accepted: 18 March 2022

Abstract
Interpretable machine learning aims at unveiling the reasons behind predictions returned by uninterpretable classifiers. One of the most valuable types of explanation consists of counterfactuals. A counterfactual explanation reveals what should have been different in an instance to observe a diverse outcome. For instance, a bank customer asks for a loan that is rejected. The counterfactual explanation consists of what should have been different for the customer in order to have the loan accepted. Recently, there has been an explosion of proposals for counterfactual explainers. The aim of this work is to survey the most recent explainers returning counterfactual explanations. We categorize explainers based on the approach adopted to return the counterfactuals, and we label them according to characteristics of the method and properties of the counterfactuals returned. In addition, we visually compare the explanations, and we report quantitative benchmarking assessing minimality, actionability, stability, diversity, discriminative power, and running time. The results make evident that the current state of the art does not provide a counterfactual explainer able to guarantee all these properties simultaneously.

Keywords Explainable AI · Counterfactual explanations · Contrastive explanations · Interpretable machine learning

1 Introduction

A widely recognized obstacle in the acceptance of Artificial Intelligence (AI) based services is the lack of interpretability (Goebel et al. 2018; Guidotti et al. 2019c; Miller 2019). Indeed, many AI systems adopt "black-box" classifiers returned by Machine

Responsible editor: Martin Atzmueller, Johannes Fürnkranz, Tomáš Kliegr, Ute Schmid.

✉ Riccardo Guidotti
riccardo.guidotti@unipi.it

[1] University of Pisa, Largo B. Pontecorvo, 3, 56127 Pisa, PI, Italy

Published online: 28 April 2022

Learning (ML) processes that map features into a class (outcome or decision) by generalizing from a dataset of examples. The cause for the lack of transparency is that the most effective classifiers, such as deep neural networks or ensemble methods, are black-box models, i.e., it is humanly impossible to understand the decision process adopted (Freitas 2013). However, explanations for a decision taken by an AI can be fundamental both for the service provider and for the users (Guidotti et al. 2019c; Miller 2019). If we consider, for instance, a bank adopting an AI to grant loans, the reasons for the loan acceptance are of interest for both the bank and for the applicant (Bhatt et al. 2020; Tomsett et al. 2018). The bank can check if the decision is sound, respectful of regulations, and in line with the desiderata of the engineers, while the applicant can decide how to react in case of rejection. However, in many cases, an explanation revealing only the *why*, i.e., the reason for a specific outcome, can be not sufficient to understand how to change the outcome.

Counterfactual explanations suggest what should be different in the input instance to change the outcome of an AI system (Lucic et al. 2020; Wachter et al. 2017). For instance, a bank customer asks for a loan that is rejected. The counterfactual explanation consists of what should have been different for the customer in order to have the loan accepted. An example of counterfactual is: "if the income would have been 1000$ higher than the current one, and if the customer had fully paid current debts with other banks, then the loan would have been accepted". Counterfactuals are at the highest level of Pearl's interpretability scale (Pearl et al. 2009), as they answer why a decision has been made by highlighting what changes in the input would lead to a different outcome. Thinking in counterfactual terms requires imagining a reality that contradicts the observed facts, hence the name "counterfactuals" (Molnar 2020). According to the cognitive psychology literature, counterfactuals help people reason on explanations that identify *cause-effect* relations (Byrne 2019; Miller 2019). The "cause" are the particular feature values of the input instance and "caused" a certain prediction, while the "effect" is the predicted outcome. In the previous example, a loan applicant may discover that her leasing would have been accepted if her income had been 1000$ higher than the current one and if she had fully paid her leasing contracts which are still open. In other words, counterfactuals are similar to how children learn through counterfactual examples (Beck et al. 2009; Buchsbaum et al. 2012), and allow to automatically explore desired "what-if" scenarios. While factual explanations aid logical reasoning (Guidotti et al. 2019a), counterfactuals add new information to what is known about the facts. Hence, they are more informative and favor creative problem solving (Byrne 2019). Moreover, they also enable actionable recourse (Karimi et al. 2021a, b).

As humans, we tend to imagine how an outcome could have been different by changing aspects that are controllable, recent, and action-based (Byrne 2019). However, a counterfactual explanation can also show that the fundamental aspects for reverting a decision are not controllable, i.e., not *actionable* (Lucic et al. 2019). For instance, in our example, we can have a counterfactual saying that the loan would have been accepted if the applicant would have been younger. Such an explanation is useless from the applicant's perspective. Indeed, while an applicant can *act* by increasing her income and by paying back her current debts, she *can not act* for becoming younger. Hence, in some applications, counterfactuals are really useful only if they show an

actionable alternative. This point is valid for a range of real applications such as leasing requests, disease prediction (Panigutti et al. 2020), job applications (Rockoff et al. 2011), university admission (Waters and Miikkulainen 2014), credit scoring (Barbaglia et al. 2020), etc. However, from another perspective, counterfactuals changing non-actionable features can be helpful to unveil biases in the decision system. Therefore, the fact that certain features are actionable and should or should not be changed depends on the "seller" of the counterfactual explainer and on the existing knowledge that the decision system does not make unfair decisions. Furthermore, besides being among the most fundamental concepts in theories of causation (Pearl et al. 2009), counterfactual explanations should also guarantee causal relationships on their own. Indeed, counterfactuals represent a causal relationship between the event that happened and its imaginary counterpart (Stepin et al. 2021). Also, since the features in a dataset are rarely independent, the imaginary counterpart, i.e., the counterfactual, should respect any known causal relations between features. Thus, a counterfactual explanation should also account for *causality*. Indeed, as humans, we are aware that when a particular aspect is changed, then another one could be required to be updated in consequence. In our example, if the counterfactual increases the length of the leasing, this could directly affect the periodic tax applied to the leasing. Thus, counterfactuals are entirely plausible only if they respect causal relationships.

As a reaction to the demand for these types of sophisticated explanations, in the last years, we have witnessed the rise of a plethora of counterfactual explanation methods, each one focusing on some desirable properties for the returned counterfactual instances (Artelt and Hammer 2019; Karimi et al. 2021a; Stepin et al. 2021; Verma et al. 2020). The aim of this work is to clarify the current panorama of counterfactual explanation methods by categorizing the various approaches with respect to the type of process adopted to retrieve the counterfactuals, the data type under analysis, and the properties guaranteed by the different counterfactuals. Moreover, we report a demonstration of various explainers with a comparison among several counterfactuals, and a quantitative numerical evaluation to measure properties such as availability, validity, minimality, actionability, plausibility, diversity, stability, discriminative power, running time, etc. We stress that concerning the evaluation of counterfactual explainers and explainability methods in general, there is no standard agreement on how to perform an objective evaluation. Thus, our benchmarking should be intended as a showcase where existing counterfactual explainers are compared with respect to different evaluation measures assessing different properties and not as absolute truth. Moreover, we highlight that the purpose of this work of survey and benchmarking is not to suggest which are the best methodologies but to detail how they work, categorize them, and present a first experimental benchmarking. Therefore, this work can be seen as a tool helping the reader to select the most appropriate set of counterfactual explainers for her problem setting. The main findings of our benchmarking show that the strategy adopted to generate counterfactual explanations markedly impacts the properties guaranteed. The majority of the explainers return examples synthetically generated through optimization strategies. These methods can be easily tuned to account for certain properties but frequently do not completely regard other ones and are also typically not efficient. On the other hand, counterfactual explainers that return examples selected from a given dataset or generated only by selecting real feature

values are typically more effective and guarantee the best trade-offs but do not achieve outstanding results for any of the desired properties. In conclusion, some approaches are empirically better than others, but the current state of the art does not provide a counterfactual explanation method able to account for all the desirable properties simultaneously.

The rest of the paper is organized as follows. Section 2 summarizes existing surveys on explainability in AI and interpretability in ML and highlights the differences between this work and previous ones. Section 3 aims at formalizing the notion of counterfactuals and existing alternative terminologies. Then, Sect. 4 presents the proposed categorization for counterfactual-based explainers. Section 5 illustrates the evaluation measures adopted in the literature to asses the goodness of counterfactual explanations, and Sect. 6 reports quantitative results employing them, as well as a practical explainer demonstration. Finally, Sect. 7 summarizes the crucial aspects that emerged from the analysis of the state of the art, and proposes future research directions.

2 Related works

In the last years emerged a widespread interest for eXplainable Artificial Intelligence (XAI) and interpretable machine learning (Goebel et al. 2018). Various books and surveys on this theme have been recently published. The book of Molnar (2020) summarizes the most widely adopted methodologies to make machine learning models interpretable, while the book of Samek et al. (2019) details how to explain deep neural networks.

In Guidotti et al. (2019c), the XAI taxonomy is based on four categories of problems with respect to the problems that explanation methods are able to solve. Similar taxonomies are presented in Arrieta et al. (2020), Bodria et al. (2021), Carvalho et al. (2019), Gilpin et al. (2018), Li et al. (2020), Murdoch et al. (2019), Samek et al. (2019). In Adadi et al. (2018), Guidotti et al. (2019c), a first distinction is between *explanation by design*, also named *intrinsic* interpretability, and *black-box* explanation, also named *post-hoc* interpretability. In Guidotti et al. (2019c), Martens et al. (2007), a further distinction classify the black-box explanation problem into model explanation, outcome explanation and black-box inspection. Model explanation, achieved by *global* explainers (Craven et al. 1995), aims at explaining the whole logic of a model. Outcome explanation, achieved by *local* explainers (Ribeiro et al. 2016; Lundberg and Lee 2017), understand the reasons for a specific outcome. Finally, another crucial distinction is between *model-specific* and *model-agnostic* explanation methods Adadi et al. (2018), Carvalho et al. (2019), Dosilovic et al. (2018), Guidotti et al. (2019c), Martens et al. (2007). The difference depends on whether the technique adopted to explain can work only on a specific black-box model or can be adopted on any black-box. In Gilpin et al. (2018) are particularly detailed the differences between the terms *explanation*, *interpretability* and *explainability*. In Arrieta et al. (2020) is presented a specific taxonomy for explainers of deep learning models. In Carvalho et al. (2019), Samek et al. (2019) we can find discussions related to a quantitative and qualitative evaluation of explanation methods. A common message among these various surveys is that the term interpretability (or transparency) is mainly used to refer a *passive* char-

acteristic of a model that makes sense for a human, while the term explainability is an *active* characteristic of a model, denoting any action taken with the intent of clarifying the decision logic. Besides, it is worth highlighting that the literature reviews related to XAI and interpretable ML are focused not just on ML and AI but also on social studies (Byrne 2019; Miller 2019), recommendation systems (Zhang and Chen 2020), model-agents (Anjomshoae et al. 2019), and domain-specific applications (Tjoa and Guan 2019).

According to the aforementioned notions, counterfactual explanations can be defined as post-hoc local explanations. Indeed, they are used to reveal the reasons for the classification of a pre-trained black-box machine learning system (hence post-hoc), and are retrieved for a specific instance under analysis (hence local). They are an active explanation method as they are not directly available on the classification model. Moreover, as we will detail in Sect. 4, in the literature exists counterfactual explanation methods which are either model specific and model agnostic. We highlight that, even if the literature frames counterfactuals as local post-hoc explanations, depending on the application, they might also be used to explain transparent approaches further, and that a set of counterfactuals might be categorized as a global explanation like in Rawal and Lakkaraju (2020).

To the best of our knowledge, at the current state of the art, there are only a few surveys specifically dedicated to counterfactual explanations. In Artelt and Hammer (2019) are reviewed model-specific methods for efficiently computing counterfactual explanations. In particular, widely adopted classification approaches are analyzed, such as SVM, logistic regressors, linear regressors, decision trees, etc., and, for every model, depending on the mathematical formulation, it is defined how a counterfactual can be identified among the records in the training set referring to existing works in the literature. However, Artelt and Hammer (2019) do not actually propose a real taxonomy or categorization for the various approaches. On the other hand, in Verma et al. (2020) existing counterfactual explanation methods are classified depending on assumptions made on the black-box, optimization alternatives, and which properties are checked/guaranteed for the counterfactuals returned. In addition, in Verma et al. (2020) are discussed the desiderata for counterfactual explanations such as validity, actionability, sparsity, data manifold closeness, and causality. Still in Verma et al. (2020), a large space is dedicated to listing open research questions involving counterfactual research. Differently from Artelt and Hammer (2019), the presentation of the methods in Verma et al. (2020) is at a very high level, and no details nor hints of how counterfactuals are returned is specified. In the book of Molnar (2020) two counterfactual explanation methods are presented in detail but without any claim of being exhaustive with respect to the state of the art. In Stepin et al. (2021) is presented an analysis of the literature review of contrastive and counterfactual explanation methods. The paper carefully describes the process for finding the inherent literature, and then analyzes the papers collected through graphics, networks, tables, and statistics. There is not a very detailed taxonomy emerging from this process. Besides, Stepin et al. (2021) focus much more on the different definitions of counterfactuals and do not put the attention on the strategies to retrieve them, which is one of the focus of this work. On the other hand, in Karimi et al. (2021a) is presented a detailed taxonomy for counterfactual explanation methods similar to the one proposed in this work.

The work of Karimi et al. (2021a) is mainly focused on algorithmic recourse, which concerns providing explanations to individuals who are unfavorably treated by automated decision-making systems. In Karimi et al. (2021a) is illustrated a high-level overview of counterfactual explanations without providing details for the many methods collected. On the contrary, in our proposal, we try to provide the main insights and algorithmic detail of every method analyzed. The short survey presented in Keane et al. (2021) addresses to which extent counterfactual explainers have been adequately evaluated and presents some of the desired properties also analyzed in this survey. Finally, in the generalist XAI survey of Bodria et al. (2021) we can find a section dedicated to counterfactual explainers, which is, however, not exhaustive. Besides, even though in Bodria et al. (2021) are tested some explanation methods, among them, there are no counterfactual explainers.

To the best of our knowledge, this manuscript is one of the few works in the literature benchmarking several counterfactual explanation methods. Indeed, non-survey papers that present a novel counterfactual explainer typically compare the proposal against Wachter et al. (2017) and with different variants of the proposed method (e.g., different loss functions). Examples of such papers are Dandl et al. (2020), Karimi et al. (2020), Mothilal et al. (2020), Van Looveren and Klaise (2021). In Mazzine and Martens (2021) is presented a benchmarking of some counterfactual explainers. However, the methods tested and the metrics adopted in this survey do not overlap with Mazzine and Martens (2021). Indeed, our aim is to test to which extent various explainers guarantee certain properties highlighted by papers in the literature in the returned explanations. On the other hand, Mazzine and Martens (2021) focuses more on proximity measures. We recommend a reading of Mazzine's benchmarking for gaining another evaluation perspective. Furthermore, also Pawelczyk et al. (2021) presents CARLA (Counterfactual And Recourse LibrAry), a python library for benchmarking counterfactual explanation methods across different datasets and different classifiers. The explainers tested in Pawelczyk et al. (2021) are a subset of those tested in the benchmark of this survey.

Thus, in this survey, we extend and complete the treatment of counterfactual explainers w.r.t. existing surveys by including recent methods and by providing an updated taxonomy that also captures different aspects that we believe are still missing in the literature. In addition, we benchmark counterfactual explainers on various datasets and black-box classifiers with respect to a set of evaluation measures that we accurately formalize.

3 Counterfactual explanations

In this section we formalize the notion of counterfactual explanation for machine learning classification. In agreement with Molnar (2020), we state that a counterfactual explanation for a prediction highlights the smallest change to the feature values that changes the prediction to a predefined output. Formally,

Definition 1 (*Counterfactual explanation*) Given a classifier b that outputs the decision $y = b(x)$ for an instance x, a counterfactual explanation consists of an instance x' such that the decision for b on x' is different from y, i.e., $b(x') \neq y$, and such that the difference between x and x' is *minimal*.

We do not formalize the concept of minimality here because, depending on the setting, it could have different meanings. We come back to this aspect at the end of the next paragraph. The classifier b is typically a black-box, i.e., a *not interpretable machine learning model* such as a neural network, an ensemble, etc. Instances x and x' consist of a set $\{x_1, x_2, \ldots, x_m\}$ of m attribute-value pairs $x_i = (a_i, v_i)$, where a_i is a feature (or attribute) and v_i is a value from the domain of a_i.[1] Therefore, counterfactual explanations, also called counterfactuals, belongs to the family of *example-based explanations* (Aamodt and Plaza 1994). Other example-based explanations are prototypes, criticisms, and influential instances (Molnar 2020). However, these instances are labeled with the same class of the instance x under analysis, i.e., $b(x) = b(x')$. Thus, none of them is able to reveal "why" $b(x) = y$ and not $b(x) \neq y$, while counterfactual explanations can. On the other hand, the not null differences between x and a counterfactual x' reveals exactly what should have been different in x for having a different outcome, i.e., $\delta_{x,x'} = \{x'_i | \forall i = 1, \ldots, m \text{ s.t. } x'_i \neq x_i\}$.

For instance, let suppose the customer x of a bank requests a loan, and the loan is rejected by the AI system of the bank based on a black-box machine learning model b. A counterfactual explanation could reveal that a hypothetical customer x' would have the loan accepted, where x' is identical to the applicant x but with a yearly income of 15,000$ instead of 12,000$, and without other debts with the bank. In this case, the hypothetical customer x' is a *counterfactual example*, and the *counterfactual explanation* $\delta_{x,x'}$ consists in the income of 15,000$ and in the lack of other debts with the bank, i.e., these minimum changes would have reverted the decision. Therefore, a counterfactual describes the dependency on the external facts that led to a particular decision made by the black-box by focusing on the differences in behavior the end-user has to make to obtain the opposite prediction w.r.t. $b(x) = y$. Going back to the *minimality*, on one scenario, it could be minimal having only one feature changed, e.g., a yearly income of 15,000$ instead of 12,000$, while on another scenario, it could be minimal having more features changed but with a smaller impact than increasing the income of 300$, e.g., no other debts and owning a car, which are binary flags. The first notion is typically referred to in the literature as *sparsity*, while the second one as *similarity* or *proximity*. Consequently, the notion of *minimality* referred to the definition of counterfactual should be formalized only when the objectives of returning counterfactuals are clear from the application. A possibility is to assign a weight to each feature estimating in this way the cost of changing it. In the following, according to the majority of the literature, we will use the term *minimality* with the meaning of *sparsity*, we will use the terms proximity or similarity otherwise.

Practically, a counterfactual explanation C, can be composed by a single counterfactual example $C = \{x'\}$, or by a set of counterfactual examples $C = \{x'_1, \ldots, x'_h\}$. We define a counterfactual explainer as a function able to return a counterfactual explanation C as follow:

[1] The domain of a feature can be continuous or categorical.

Definition 2 (*Counterfactual explainer*) A *counterfactual explainer* is a function f_k that takes as input a classifier b, a set X of known instances, and a given instance of interest x, and with its application $C = f_k(x, b, X)$ returns a set $C = \{x'_1, \ldots, x'_h\}$ of $h \leq k$ of *valid* counterfactual examples where k is the number of counterfactuals required.

Most of the counterfactual explainers in the literature are designed as f_1 function to return a single valid counterfactual. If $C = \emptyset$, i.e., $h = 0$, it means that the explainer was not able to find any valid counterfactual.

In the following we illustrate properties of counterfactual explanations and counterfactual explainers shared by the various papers in the literature.

3.1 Proprieties of counterfactual explanations

Research in counterfactual explanations has focused on addressing the problem of finding counterfactual examples guaranteeing some desirable properties. In the following, we formalize the most widely used and shared desirable properties, i.e., validity, minimality, similarity, plausibility, discriminative power, actionability, causality, and diversity. The most rigorous and inspiring works in this direction are Mothilal et al. (2020), Verma et al. (2020).

- *Validity.* A counterfactual x' is valid iff it actually changes the classification outcome with respect to the original one, i.e., $b(x') \neq b(x)$.
- *Minimality (Sparsity).* There should not be any other valid counterfactual example x'' such that the number of different attribute value pairs between x and x' is higher than the number of different attribute value pairs between x and x''. We say that x' is minimal iif $\nexists\, x''$ s.t. $|\delta_{x,x''}| < |\delta_{x,x'}|$, where $|A|$ returns the size of set A.
- *Similarity.* A counterfactual x' should be similar to x, i.e., given a distance function d in the domain of x, the distance between x and x' should be as small as possible $d(x, x') < \varepsilon$, where ε is a predefined maximum distance threshold. Similarity is often referred to as *proximity*.
- *Plausibility.* Given a reference population X, a counterfactual x' is plausible if the feature values in x' are coherent with those in X. This practically means that the feature values of x' should not be higher/smaller than those observable in X, and that x' should not be labeled as an outlier with respect to the instances in X. Plausibility helps in increasing trust towards the explanation: it would be hard to trust a counterfactual if it is a combination of features that are unrealistic with respect to existing examples. On the contrary, a plausible counterfactual is "realistic" because it is "similar" to the known dataset and adheres to observed correlations among the features. Plausibility is also named *feasibility* or *reliability*. Various approaches are being proposed to check for plausibility. Laugel et al. (2019) proposes to check that a counterfactual is plausible (justified in the paper) through a concept of ϵ-chain distance with respect to a real record in X. Artelt and Hammer (2020a) suggests adding specific constraints to control and measure plausibility in terms of density, i.e., a plausible counterfactual x must lie in a dense area with respect to the instances in X. In Artelt et al. (2021) are presented evalu-

ation measures showing that plausibility also helps for robustness and stability of counterfactual explanations.

- *Discriminative. Power* A counterfactual x' should show a high discriminative power for recognizing the reasons for the decision outcome Guidotti et al. (2019b), Kim et al. (2016), Mothilal et al. (2020). Indeed, being a counterfactual an explanation, it must help in figuring out why a different output can be obtained with x'. In other words, looking at x and x', also as humans we would have classified $b(x) = y$ and $b(x') \neq y$ due to the differences $\delta_{x,x'}$ between x and x'. We highlight that, with respect to the above definition and the literature cited above, the discriminative power is defined based on a subjective basis that can be difficult to quantify without experiments involving humans. As illustrated in Sect. 5, this issue is typically addressed by relying on simple decision models that are supposed to approximate human behavior.
- *Actionability.* Given a set A of *actionable features*, i.e., features that can be mutated, a counterfactual x' is actionable iif all the differences between x and x' refers only to actionable features, i.e., $\nexists\ a_i \in \delta_{x,x'}$ s.t. $x'_i = (a_i, v_i) \wedge x_i \notin A$. Examples of non-actionable features that cannot be changed in a counterfactual are age, gender, race, etc. Indeed, a counterfactual should never change the non-actionable (immutable) features. Actionability is also referred to as *feasibility*. In Ustun et al. (2019) is used the term *recourse* to indicate a counterfactual that accounts for the actionability of the features changed. It is important to underline that the above formalization of actionability is a soft one and could not be sufficient if the infeasibility of a counterfactual comes from a combination of different factors, i.e., living in a particular place and having a particular job that is impossible due to the contextual circumstances. However, to the best of our knowledge, no one of the papers reported in this survey considers this complex notion of actionability.
- *Causality.* Let G be a Directed Acyclic Graph (DAG) where every node models a feature and there is a directed edge from i to j if i contributes in causing j. The DAG G describes the known causalities among features. Thus, given a DAG G, a counterfactual x' respects the causalities in G iif $\forall x'_i = (a_i, v_i) \in \delta_{x,x'}$ such that the node i in G has at least an incoming/outcoming edge, the value v_i maintains any known causal relation between i and the values $v_{j_1}, \ldots, v_{j_m}$, where the features $j_1, \ldots, j_m$ identifies the nodes connected with i in G. Indeed, in order to be really plausible and actionable, a counterfactual should maintain any known causal relationship between features. For instance, increasing the number of years for a loan typically implies to increase also the interest rate. We highlight that this notion of causality is not the same causality captured by counterfactuals. Indeed, counterfactuals model Pearl causality between input and outcome, while the desired property discussed here is among features, and it is also connected to plausibility.
- *Diversity.* Let $C = \{x'_1, \ldots, x'_k\}$ be a set of k (valid) counterfactuals for the instance x. The counterfactual explanation C should be formed by diverse counterfactuals, i.e., while every counterfactual $x'_i \in C$ should be minimal and similar to x, the difference among all the counterfactuals in C should be maximized (Mothilal et al. 2020; Tsirtsis and Rodriguez 2020). For instance, three (similar) counterfactuals saying that a yearly income of 15,000$, of 15,100$, and of 14,800$ is going to

change the outcome are less useful than three (different) counterfactuals saying that the outcome can be changed (i) with a yearly income of 15,000$, (ii) by owning a car, or (iii) by first paying back the other debts. Indeed, with the second set of diverse counterfactuals, there are more possible actions to change the classification outcome.

In addition, we can say that a counterfactual explanation is not *available* if the explanation method failed to find it.

The level of satisfaction of these properties by a certain counterfactual example x', or set of counterfactual examples C, can be used to measure the goodness of an explanation as detailed in Sect. 5.

3.2 Properties of counterfactual explainers

Besides, research in counterfactual explanations (Bodria et al. 2021; Guidotti and Ruggieri 2019) aimed at guaranteeing some desirable properties for the counterfactual explainers, i.e., efficiency, stability, and fairness.

- *Efficiency.* An explainer f should return the set C of counterfactuals fast enough to ensure that they can be used in real life applications.
- *Stability.* Given two similar instances x_1 and x_2 obtaining the same classification from the classifier b, i.e., $y = b(x_1) = b(x_2)$, then an explainer f should return two similar set C_1, C_2 of counterfactuals. Thus, the counterfactual explainer f should show *stability* across various explanations such that similar instances would receive similar explanations. Stability is often referred to as *robustness*.
- *Fairness.* A counterfactual explainer is fair if, given a record x, any counterfactual explanation x' for x is valid both in the "actual world" and in the "counterfactual world" when in x' can also be applied changes leading it to belong to a different demographic group. For instance, suppose that x' gets the loan accepted by reducing its duration to 10 years from 15 years. The explainer is fair if x' has the loan accepted, i.e., is still a valid counterfactual, also changing, e.g., the ethnicity. Various scenarios of counterfactual fair explanations are described in detail in Kusner et al. (2017) and in Von Kügelgen et al. (2020). The features that can be changed to check the fairness largely correspond to the non-actionable ones.

3.3 Related terms and definitions

In the literature, terms different from counterfactual have been used to indicate a similar concept or a highly related one, i.e., contrastive explanations, adversarial learning, exemplars, prototypes, criticism, influential instances, and inverse classification. In the following we analyze these terms and their relationships with counterfactual explanations.

Contrastive explanations (Lipton 1990; Miller 2018) are recognized to be in the form "$b(x) = y$ instead of $b(x) = y'$ because features $x_{i_1}, \ldots, x_{i_m}$ have values $v_{i_1}, \ldots, v_{i_m}$ instead of values $v'_{i_1}, \ldots, v'_{i_m}$ ". Thus, it is the ability to distinguish the answer to an explanatory question from a set of contrastive hypothesized alterna-

tives that provides the user sufficient information to unveil the reasoning behind the decision. In McGill et al. (1993), Stepin et al. (2021) is illustrated that contrastive explanations are related to situations where different outcomes are analyzed: "what made the difference between the customer who got the loan accepted and the customer who got the loan refused?". On the other hand, counterfactual reasoning is claimed to deal with cases where the antecedent, i.e., the instance under analysis, is varied to change the prediction outcome: "would the customer have had the loan accepted if she had an income of 15,000$?". In Dhurandhar et al. (2018), the features which are minimally sufficient to obtain a certain outcome are called *pertinent positives*, while the features whose absence is necessary for the outcome are named *pertinent negatives*. Hence, pertinent negatives are indeed contrastive explanations, and in turn, they are strictly related to counterfactual explanations. In our opinion, practically speaking, in XAI applications, there is no difference between counterfactual and contrastive explanations. Indeed, in both cases, the aim is to find what would have changed the decision, either altering x or by comparing x with another instance.

On the other hand, *adversarial learning* is a closely related area to counterfactual search, but the two terms have not the same meaning. Indeed, adversarial examples are not aimed to pursue the same goal of counterfactual explanations. Adversarial learning attempts to fool models by supplying deceptive input (Ballet et al. 2019; Kianpour and Wen 2019). The idea here is to generate the minimum number of changes in a given input in order to classify it differently but with the objective of discovering highly-confident misclassification examples. While the strategies used to find the adversarial examples can be the same adopted to retrieve counterfactuals, the objectives are different. For example, adversarial learning applied to images aims at finding a humanly imperceptible change in the input image that changes the prediction outcome with the idea of fooling the classifier. On the contrary, counterfactual explainers applied to images aim at highlighting significant parts of the image that changes the prediction outcome with the idea of explaining the classifier. Furthermore, properties such as plausibility, actionability, and causality are hardly taken into account by adversarial learning approaches.

To complete the treatment, since counterfactual explanations belong to the family of example-based explanations (Molnar 2020), it is important to mention the counter-part of counterfactual explanations, i.e., the *exemplars* or *prototypes*. A prototype $\tilde{x}$, also called archetype or artifact, is an object representing a set of similar records that obtains the same classification of x, i.e., $b(x) = b(\tilde{x})$. A prototype should clarify to the user the aspect leading to a specific outcome (Bien et al. 2011). A prototype can be a record from the training dataset close x, the centroid of the cluster to which x belongs, or a record synthetically generated. In Kim et al. (2016) is also defined the notion of *criticism* as an exemplar instance that is not well represented through prototypes. The purpose of criticisms is to provide insights into the characteristics/discriminative aspects of data points that prototypes do not represent well. We highlight that prototypes and criticisms can also be used independently from a machine learning model to describe the data. Besides, *influential instances* are exemplar instances whose removal has a strong effect on the trained model (Koh et al. 2017).

Finally, the problem of finding counterfactual examples in Aggarwal et al. (2010), Lash et al. (2017b) is named *inverse classification* as the aim is on perturbing the input

to change the predicted outcome. Therefore, the counterfactual explainer f can be considered an inverse classifier of b.

4 How to find counterfactual explanations

We categorize counterfactual explanation methods with respect to the strategy adopted to retrieve the exemplars, and we further describe them according to certain functional characteristics and properties guaranteed for counterfactuals. Our objective is to provide to the reader a guide to map a set of desiderata and requirements of the user with a set of compatible counterfactual explanation methods. Table 1 summarizes state-of-the-art counterfactual explanation methods and annotates them with respect to the strategies, characteristics, and properties described in the next section. The explainers are sorted first with respect to the strategy and then chronologically. For each method, we provide a link to the source code or library (if available).

Counterfactual explanations can be exploited to interpret the decisions returned by AI systems employed in various settings. In particular, in the literature are recognized the following problems where counterfactual explainers can be used (Stepin et al. 2021): classification, regression, knowledge engineering, planning, and recommendation. In this survey we focus the analysis and categorization on counterfactual explanation methods designed to explain black-box classification models because the large majority of the papers in the literature give attention to this problem.

Since the research field of counterfactual explanations is emerging and the terms used are various, as discussed in the previous section, it is practically impossible to perform a systematic literature review with keywords. Therefore, we started from a set of *core* papers officially published in conference proceedings and recognized as important by the research community due to the high number of citations. Then, we collected papers using backward/forward-search from the cited/citing references. We repeated the procedure up to two hops. Also, we researched papers on Google and Google Scholar search engines using the keywords retrieved from the core papers. We decided to focus on papers about counterfactual explanations that had been officially published in journals and conference proceedings, on unpublished papers with a no negligible number of citations, and on those offering a well-documented library.

4.1 Categorization of counterfactual explainers

The first aspect that we adopt to distinguish the various explainers is related to "how" they retrieve the counterfactual explanations. In the literature we mainly observe the following strategies:

- *Optimization (OPT).* Counterfactual explainers based on optimization strategies defines a loss function that accounts for desired properties and adopts existing optimization algorithms to minimize it.
- *Heuristic Search Strategy (HSS).* Counterfactual explainers based on heuristic search strategies aim at finding counterfactuals through local and heuristic choices that at each iteration minimize a certain cost function.

Table 1 Taxonomy

Name	References	Strategy	Model agnostic	Data agnostic	Categorical	Validity	Actionability	Causality	Exogenous	Multiple	Code
BF	–	BF	✓	TAB		✓			✓	✓	link
OAE	Cui et al. (2015)	OPT	ENS	TAB	✓	✓			✓		–
WACH	Wachter et al. (2017)	OPT	DIF	TAB		✓			✓		link
CEM	Dhurandhar et al. (2018)	OPT	DIF	✓		✓			✓		link
ICACE	Mc Grath et al. (2018)	OPT	DIF	TAB		✓			✓		–
CEML	Artelt (2019)	OPT	✓	TAB	✓	✓	✓		✓		link
EMAP	Chapman-Rounds et al. (2019)	OPT	✓	✓	✓	✓			✓	✓	–
MACEM	Dhurandhar et al. (2019)	OPT	✓	TAB	✓	✓			✓		–
CRUDS	Downs et al. (2020)	OPT	DIF	TAB	✓	✓		✓	✓		–
REVISE	Joshi et al. (2019)	OPT	DIF	TAB	✓	✓		✓	✓		–
FOCUS	Lucic et al. (2019)	OPT	ENS	TAB		✓			✓		link
EBCF	Mahajan et al. (2019)	OPT	DIF	TAB	✓	✓		✓	✓		link
DCE	Russell (2019)	OPT	LIN	TAB	✓	✓			✓	✓	link
ACTREC	Ustun et al. (2019)	OPT	DIF	TAB	✓	✓	✓		✓	✓	link
DACE	Kanamori et al. (2020)	OPT	LIN/ENS	TAB	✓	✓	✓		✓		link
MACE	Karimi et al. (2020)	OPT	✓	TAB	✓	✓	✓		✓	✓	link
DICE	Mothilal et al. (2020)	OPT	DIF	TAB	✓	✓	✓		✓	✓	link
C- CHAVE	Pawelczyk et al. (2020)	OPT	✓	TAB	✓	✓	✓		✓	✓	link
SYNTH	Ramakrishnan et al. (2020)	OPT	DIF	TAB	✓	✓			✓		link
ARES	Rawal and Lakkaraju (2020)	OPT	✓	TAB		✓	✓		✓	✓	–

Table 1 continued

Name	References	Strategy	Model agnostic	Data agnostic	Categorical	Validity	Actionability	Causality	Exogenous	Multiple	Code
SCOUT	Wang and Vasconcelos (2020)	OPT	CNN	IMG		✓			✓		–
FRACE	Zhao (2020)	OPT	DIF	IMG		✓			✓		–
CEODT	Carreira-Perpiñán and Hada (2021)	OPT	TREE	✓	✓	✓			✓		link
DECE	Cheng et al. (2021)	OPT	DIF	TAB		✓	✓		✓		link
SGNCE	Mohammadi et al. (2021)	OPT	DIF	TAB	✓	✓	✓		✓		–
OCEAN	Parmentier and Vidal (2021)	OPT	ENS	TAB	✓	✓	✓		✓		link
ORDCE	Kanamori et al. (2021)	OPT	✓	TAB	✓	✓	✓	✓	✓		link
ALGREC	Karimi et al. (2021b)	OPT	DIF	TAB		✓	✓	✓	✓		link
PIECE	Kenny and Keane (2021)	OPT	CNN	IMG		✓			✓		link
CEGP	Van Looveren and Klaise (2021)	OPT	DIF	✓	✓	✓			✓		link
POLYJUICE	Wu et al. (2021)	OPT	✓	TXT						✓	link
SEDC	Martens and Provost (2014)	HSS	✓	TXT		✓			✓	✓	link
GIC	Lash et al. (2017b)	HSS	✓	TAB			✓	✓	✓		link
GSG	Laugel et al. (2018)	HSS	✓	✓		✓			✓		link
POLARIS	Zhang et al. (2018)	HSS	DIF	✓		✓			✓	✓	–
CVE	Goyal et al. (2019)	HSS	✓	IMG		✓			✓		–
CADEX	Moore et al. (2019)	HSS	DIF	TAB	✓			✓			link
CFSHAP	Rathi (2019)	HSS	✓	TAB		✓			✓		–
CERTIFAI	Sharma et al. (2019)	HSS	✓	✓	✓	✓			✓	✓	link
PCATTGAN	Arrieta and Ser (2020)	HSS	DIF	IMG		✓			✓	✓	–
MOC	Dandl et al. (2020)	HSS	✓	TAB	✓	✓	✓		✓	✓	link

Table 1 continued

Name	References	Strategy	Model agnostic	Data agnostic	Categorical	Validity	Actionability	Causality	Exogenous	Multiple	Code
VICE	Gomez et al. (2020)	HSS	✓	TAB	✓	✓	✓		✓		link
PERMUTEATTACK	Hashemi and Fathi (2020)	HSS	✓	TAB	✓	✓			✓	✓	–
GRACON	Kang et al. (2020)	HSS	DIF	✓		✓			✓		–
GRACE	Le et al. (2020)	HSS	DIF	TAB	✓	✓			✓		–
MCBRP	Lucic et al. (2020)	HSS	✓	TAB		✓				✓	link
LIME- C/SHAP- C	Ramon et al. (2020)	HSS	✓	✓		✓			✓	✓	l1,l2
CLEAR	White and d'Avila Garcez (2020)	HSS	✓	TAB	✓	✓			✓	✓	link
PCIG	Yang et al. (2020)	HSS	DIF	TXT		✓			✓	✓	–
GECO	Schleich et al. (2021)	HSS	✓	TAB	✓	✓	✓		✓	✓	link
SEDCT	Vermeire and Martens (2022)	HSS	✓	IMG		✓			✓	✓	–
NNCE	Wexler et al. (2020)	IB	✓	TAB	✓	✓	✓	✓		✓	link
CBCE	Keane and Smyth (2020)	IB	✓	TAB		✓				✓	–
FACE	Poyiadzi et al. (2020)	IB	✓	✓	✓	✓	✓			✓	–
NICE	Brughmans and Martens (2021)	IB	✓	TAB	✓	✓				✓	–
TBCE	Craven et al. (1995)	DT	✓	TAB	✓	✓	✓		✓	✓	–
FT	Tolomei et al. (2017)	DT	ENS	TAB	✓	✓	✓			✓	link
LORE	Guidotti et al. (2019a)	DT	✓	✓	✓	✓	✓		✓	✓	link
FOILTREE	Van Der Waa et al. (2019)	DT	✓	TAB					✓	✓	–
RF- OCSE	Fernández et al. (2020)	DT	ENS	TAB	✓	✓			✓	✓	link

Strategy adopted: Brute Force (BF), Optimization (OPT), Heuristic Search Strategy (HSS), Instance-Based (IB), Decision Tree based (DT). If it is not model agnostic, then it is able to explain one of these classifier: Differentiable (DIF), linear (LIN), Tree-base Ensembles (ENS), Decision Trees (DT). If it is not data agnostic, it is able to process Tabular (TAB), Images (IMG), or Text (TXT) data type. A tick occurs if it can handle categorical features, actionability, causality, if it guarantees validity, if it is exogenous, and if it returns multiple counterfactuals

– *Instance-Based (IB).* Instance-based counterfactual explainers retrieve counterfactuals by selecting the most similar examples from a dataset.
– *Decision Tree (DT).* Counterfactual explainers based on decision trees approximate the behavior of the black-box with a decision tree and then exploit the tree structure to identify counterfactual explanations.

We underline that, from a certain perspective, IB strategies and DT strategies can be seen as sub-categories of HSS strategies. However, we preferred to keep them separated to focus on these specific types of methodologies. In the remaining of this section, we describe in detail the counterfactual explainers reported in Table 1 separating them into subsections with respect to the aforementioned classification. We also add a subsection considering alternative solutions to retrieve counterfactual explanations and problem settings with different models to explain, e.g., recommender systems instead of classifiers.

Most of the counterfactual explanation methods are *local post-hoc explainers*. However, according to the literature in XAI (Adadi et al. 2018; Guidotti et al. 2019c), they can be further distinguished between:

– *Model Agnostic* if the explainer can be employed to explain any black-box.
– *Model Specific* if it can be employed only on a specific black-box model.

In Table 1 we use a check mark (✓) if the explainer is *model agnostic*, otherwise we specify the black-box type. Since many counterfactual explainers are specifically designed for "differentiable" black-boxes such as neural network, we adopt the acronym DIF to identify this family of models.

Another categorization is relative to the fact that can be used on any data type or only on specific data types. In Table 1 we use ✓ if the explainer is *data agnostic*, otherwise we report the acronym of the data type if it is data specific: TAB for tabular data, IMG for images, TXT for text. Moreover, for methods working on tabular data, it is important to know if the method is able to handle categorical attributes such as sex, ethnicity, color, etc. Indeed, most of the explainers work through numerical matrices and require a notion of distance between points, and it is not necessarily granted the fact that the methods are able to appropriately manage categorical attributes. If a method can handle categorical attributes the *categorical* column is annotated with ✓.

Depending on the data type and problem considered, counterfactual explanations can be represented in different forms (Stepin et al. 2021). For tabular data, they can be represented as values (numbers or intervals) whose transformation changes the output of the black-box model, i.e., the values contained in $\delta_{x,x'}$. However, the same information can be represented through linguistic expression with sentences storytelling the content of $\delta_{x,x'}$. For images, counterfactuals are specific regions that should be varied in the input image to alterate the output. The content of these updated regions is formally modeled with $\delta_{x,x'}$. We underline that this *is different to saliency maps* where the pixels highlighted are those responsible for the outcome (Bodria et al. 2021). However, the saliency maps defined by Guidotti et al. (2019b) also offer this counterfactual contribution. For textual data, we can have a similar representation to the one described for tabular data with linguistic expressions.

Also, in line with the properties illustrated in Sect. 3.1, we highlight with check marks explanation methods which *guarantee validity*, are able to *handle actionabil-*

ity and which are able to *handle causality*.[2] On the other hand, plausibility is not explicitly guaranteed by methods unless they are *endogenous* counterfactual explanation methods. Indeed, another distinction of counterfactual explainers is between solutions that generate synthetic counterfactuals, versus those that try to find them in a dataset (Keane and Smyth 2020).

- *Endogenous* explainers return examples either selected from a given dataset, or generated only by selecting feature values from the given dataset.
- *Exogenous* explainers return examples generated without guaranteeing the presence of naturally occurring feature values. They rely on instances obtained through interpolation and/or random data generation.

In Table 1 we observe that the majority of explainers in the literature produce exogenous counterfactuals. However, endogenous counterfactuals naturally guarantee the plausibility of counterfactual explanations. We do not include in Table 1 a column indicating plausibility because, except for the endogenous explainers, no methods can guarantee plausibility. Indeed, as shown in the following, having a penalization term in the loss function to control plausibility does not guarantee it. We underline that, with respect to Definition 2, both endogenous and exogenous counterfactual explainers always require a reference set X, either to select the instances to return or to implement procedures to generate synthetic instances. Finally, we add a check-mark for methods returning more than a counterfactual explanation (*multiple* column).

4.2 Finding counterfactuals with a brute force procedure

Before presenting the strategies most widely adopted in the literature by counterfactual explanation methods, it is worth mentioning that such counterfactual examples can be found through a *Brute Force (BF)* procedure (BF). A BF procedure can find counterfactuals for any black-box classifier through a sort of "grid search" among the features describing the data with specified step size and for a selected range of values.

More in details, BF generates all the possible variations of x with respect to any of the subsets in F by replacing a feature value in x with any representative value of a bin of the feature, where F is the set of features describing x. The weakness of this approach is that it has a high computational complexity tied to the number of features $m = |F|$, to the range of values, and to the step size used to search for counterfactuals. In particular, the complexity of BF is $O(\binom{|F|}{m'} \cdot m \cdot r)$, where m' is the maximum number of features to vary simultaneously and r is the maximum number of values to test. The complexity can be mitigated by setting low values for m' and r, i.e., not considering many simultaneous changes and few alternative values but substantially reducing the search space and discarding potential "optimus" counterfactuals. The greater are m' and r, the larger the search space explored by BF. This implies a larger number of counterfactuals but also a higher complexity. For these reasons, BF is rarely applied. BF can be constrained to work only on actionable features by replacing F with $A \subseteq F$ where A is the set of actionable features. Also, this operation can reduce the complexity if $|A| \ll |F|$.

[2] Causality is perhaps the less studied property in the literature, therefore the treatment with respect to this property in this survey is limited if compared with the others.

An implementation of BF is offered by the *FAT* library.[3] The implementation is model agnostic, works only on tabular datasets, guarantees validity and returns more than one counterfactual ($k \geq 1$), even though it is not possible to specify how many of them. It is considered an exogenous counterfactual explainer because the values used in x' to change to outcome are in the domain of x but are not exactly the values observed in X. An alternative of the BF approach that works in a similar fashion but with a markedly lower computational complexity is a completely pure random (RCE) approach that randomly selects the features to vary and the values to replace and returns a counterfactual if it is valid. The RCE approach has not any guarantee of optimality. Indeed, RCE could return a counterfactual x' with ten features changed with respect x when only two would have been enough.

4.3 Finding counterfactuals by solving optimization problems

Most of the counterfactual explainers in the literature (see Table 1) return counterfactuals by solving an optimization problem. The problem is typically designed through the definition of a loss function aimed at guaranteeing a set of desired properties for the counterfactuals returned. The loss typically takes as input the instance under analysis x, a candidate counterfactual x' and the desired (counterfactual) outcome $y' \neq b(x)$. In this setting, the objective is to find a counterfactual instance that minimizes this loss using an optimization (OPT) algorithm. Each method in the literature that adopt the OPT strategy accounts for slightly different aspects by using variations of the loss function. In the following, we describe some peculiarities of the most widely adopted and cited methods solving an optimization problem to retrieve counterfactuals.

OAE. The first counterfactual explainer based on optimization has been proposed by Cui et al. (2015), but without using the term counterfactual. The fact that the method of Optimal Action Extraction (OAE) is less known than WACH is probably due to the fact that OAE is a model-specific approach for additive models, i.e., ensembles. The idea of the paper is to model the trees composing the ensemble through logical formulae and then solve an Integer Linear Programming problem with the IBM ILOG CPLEX solver.

WACH. Wachter et al. (2017) is among the first paper to propose a counterfactual explainer, and probably is the most famous one.[4] The loss function minimized by WACH is defined as

$$\lambda(b(x') - y')^2 + d(x, x')$$

where the first term is the quadratic distance between the desired outcome y' and the classifier prediction on x', and the second term is the distance d between x and x'. The parameter λ balances the contribution of the first term against the second term. A low value of λ means that we prefer x' similar to x, while a high value of λ means that we aim for predictions close to the desired outcome y'. In Wachter et al. (2017)

[3] https://fat-forensics.org/.

[4] In Table 1 we link a third party implementation as Wachter et al. (2017) do not make available any usable version.

is suggested to maximize λ while minimizing the loss. Thus, λ becomes a parameter of the search problem and a tolerance ϵ is used to constraint the classification of x' not too far away from y', i.e., $|b(x') - y'| < \epsilon$. Hence, the loss measures how far the outcome of the counterfactual $b(x')$ is from the desired outcome y', and how far the counterfactual x' is from the instance of interest x'.

The distance function d adopted is a crucial characteristic in any counterfactual explainer. Wachter et al. adopts the Manhattan distance weighted with the inverse median absolute deviation (MAD) of each feature, i.e.,

$$d(x, x') = \sum_{i}^{m} \frac{|x_i - x'_i|}{MAD_i}$$

where MAD_i is the median absolute deviation of the i-th feature. Any other distance, such as the Euclidean distance, can theoretically be used. What is important, is that either the dataset is normalized a priori (and this is often the case for black-boxes such as DNN or SVM), or the differences between the features are normalized during the calculus with a strategy similar to MAD.

The loss function can be minimized through any suitable optimization algorithm, such as the Nelder-Mead (Simplex) method (Powell 1973). If the black-box is differentiable and it is possible to access the gradient, then gradient-based methods like Adam (Kingma and Ba 2015) can be used. In summary, the WACH method works as follows. Given an instance x to be explained, a black-box b, and the desired outcome y', WACH randomly initializes x' and λ. Then it optimizes the loss updating the values in x'. If the constraint $|b(x') - y'| < \epsilon$ is not respected, then λ is increased and another optimization step is run until $|b(x') - y'| < \epsilon$ is verified. A drawback of WACH is that setting the initial value of λ a priori is unclear as well as the value of ϵ. WACH is an exogenous explainer specifically designed to explain differentiable classifiers acting on tabular data. It returns a single counterfactual. It does not handle categorical features, nor actionability, or causality and does not account for plausibility.

Mc Grath et al. (2018) extends WACH to explain credit application predictions. Mc Grath et al. introduce a weight vector to the distance metric to prefer counterfactuals acting on highly discriminative features. Two strategies are proposed to generate these weight vectors. The first one relies on the global feature importance using analysis of variance (ANOVA F-values) between each feature and the target. The second one relies on a Nearest Neighbors approach aggregating over the relative changes in the neighborhood with respect to x.

CEM. The contrastive explanation method (CEM) described in Dhurandhar et al. (2018) is based on the notions of *pertinent positives* and *pertinent negatives* described in Sect. 3.3. Dhurandhar et al. defines a counterfactual x' as $x' = x + \delta$ where δ is a perturbation applied to x such that $b(x + \delta) \neq b(x)$. CEM ensures that the modified record x' is plausible through an autoencoder that evaluates the closeness of x' to known data. In particular, two different (but similar) loss functions are defined to retrieve pertinent positives and pertinent negatives. For the purpose of this survey we focus on the loss used to find pertinent negatives that corresponds to the contrastive

explanations:

$$\alpha f(b(x), b(x+\delta)) + \beta \|\delta\|_1 + \|\delta\|_2^2 + \gamma \|\delta - AE(x+\delta)\|_2^2$$

where f encourages $x + \delta$ to be predicted as a different class than x, the second and third terms are jointly called the elastic net regularizer (Zou and Hastie 2005), and the last term is an L2 reconstruction error of x' evaluated by an autoencoder, with $AE(x)$ denoting the reconstructed example of x using an autoencoder, and α, β, γ being the regularization coefficients. Therefore, CEM return a single counterfactual as a result of the optimization of FISTA (Beck and Teboulle 2009) of the aforementioned loss function. Being model agnostic, CEM is experimented on different data types but only on differentiable classifiers. Like other explainers based on optimization, it returns exogenous counterfactuals. CEM guarantees validity but cannot handle categorical attributes, actionability, and causality. In the available implementation, actionability can be practically managed by constraining immutable features in the values of x.

CEML. CEML is a toolbox for computing counterfactuals Artelt (2019) not formally presented in any paper. However, it is based on an optimization approach, and the library offers different solutions for different types of black-box models to be explained. The interested reader can refer to the following papers pointed by the documentation Artelt and Hammer (2019, 2020a, b).

EMAP. Chapman-Rounds et al. (2019) present EMAP, Explanation by Minimal Adversarial Perturbation. EMAP is named FIMAP in Chapman-Rounds et al. (2021) standing for Feature Importance by Minimal Adversarial Perturbation. EMAP/FIMAP is a model and data-agnostic approach returning counterfactual explanation following the idea of adversarial perturbations, i.e., minimality is highly preferred while plausibility is not considered at all. EMAP/FIMAP trains a surrogate neural network model s to approximate the behavior of the black-box b in order to have access to the gradient of the computation. Then, it searches for the optimal parameter setting θ using a standard gradient-based model. The parameters θ are used by a differentiable function g, i.e., another neural network responsible for returning the minimal perturbation to be applied to x to obtain $b(x) \neq y$. EMAP/FIMAP is extended to work also for categorical data and mixed data types.

MACEM. In Dhurandhar et al. (2019), CEM is extended with the Model Agnostic Contrastive Explanations Method (MACEM). MACEM gains the model agnostic property by using a function that estimates the gradient instead of directly calculating it. Thus, while FOCUS approximate the classifier, MACEM approximates directly the gradient. The gradient estimation is performed through a function that randomly select q different and independent random directions and than computes the approximated gradient as the averaged difference between $(b(x + \alpha u_j) - b(x))/\alpha u_j$ where $\alpha > 0$ is a smoothing parameter and u_j is a random direction. MACEM also includes the treatment of categorical features through the Frequency Map Approach or the Simplex Sampling Approach. Finally, also MACEM is based on the FISTA optimizer.

CRUDS. CRUDS (Downs et al. 2020) can be seen as an extension of REVISE that uses Conditional Subspace VAE (CSVAE) (Klys et al. 2018) instead of VAE. Shortly, CRUDS first learns a latent subspace using the CSVAE predictive of the outcome y. Then, it generates counterfactuals similar to REVISE by changing only relevant latent

features. After that, when user constraints are given, or causal knowledge is available, CRUDS filters out bad counterfactuals. Finally, it summarizes counterfactuals for interpretability.

REVISE. Joshi et al. (2019) present REVISE, a counterfactual explainer accounting for actionability and causality. Similarly to EBCF, also REVISE works on the latent space of a trained Variational AutoEncoder (VAE) to find the counterfactuals through an optimization algorithm. At each iteration, the candidate instance x' is updated by minimizing a loss function that accounts for the desired class y' and for the distance between x and x'. The loss of REVISE is extended to account for known causalities in the cross-entropy function, while actionability is obtained by defining immutable variables which are not allowed to change to find the recourse. We highlight that also immutable features can be confounding variables for other actionable features.

FOCUS. In Lucic et al. (2019) the authors present FOCUS, Flexible Optimizable Counterfactual Explanations for Tree Ensembles. FOCUS adopts an optimization strategy to find counterfactuals and is mainly extended to be applied to (non-differentiable) tree ensembles. This goal is reached by using probabilistic model approximations in the optimization framework. More in detail, the term $(b(x') - y')^2$ in WACH is replaced with $\mathbb{1}_{y'=b(x')}\tilde{b}(x')$ where b is a tree ensemble, and $\tilde{b}$ is a differentiable approximation of b. In particular, $\tilde{b}$ is obtained through differentiable approximations of decision trees using an activation function in each internal node. FOCUS is tested with Euclidean, Cosine, Manhattan, and Mahalanobis distance metrics for the second term of the loss. As optimizer, similarly to WACH, it adopts Adam. FOCUS is a model-specific explainer developed for tree ensembles. However, it is theoretically model-agnostic. Also all the other properties are inherited from WACH.

EBCF. The Example-Based CounterFactual explainer (EBCF) presented in Mahajan et al. (2019) includes a variational autoencoder (VAE) that regularizes the generation of counterfactuals, and a fine-tuning phase that model parameters to support feasibility through causality. EBCF adds to the loss function a regularization term in the form of a KL divergence between the prior distribution of having x' given the class y', and the prior for the encoder of having x' given x and y'. Additionally, it accounts for plausibility by checking that known causal relationships are respected. To this aim, the VAE is fine-tuned with candidate counterfactuals respecting and not respecting the causal relationship. EBCF adopts the Adam optimizer, handles categorical features with one-hot encoded vectors, and controls their feasibility through the VAE.

DCE. The efficient search for Diverse Coherent Explanations (DCE) proposed by Russell (2019) extends WACH with the aim of finding *coherent* and *different* counterfactuals. *Coherent* means that solutions are guaranteed to map back onto the underlying data structure, i.e., they are plausible, while *diverse* means that unveil different reasons for a given outcome. Russell formulates the problem as a linear program with b being a linear classifier and with a distance function d that takes the form of a weighted L1 norm. Features are treated as integer constraints through a one-hot encoding for both continuous and categorical values, and the formulation is solved using mixed integer programming. The problem with this encoding is that the extra degrees of freedom allow implausible values (for example, by turning all indicator variables on). A set of linear constraints guarantee that the counterfactual found is plausible. Similarly,

diversity is induced through additional constraints that reduce the possible values with respect to the counterfactuals already generated.

ACTREC. Ustun et al. (2019) are among the firsts to address the problem of actionability in counterfactual explanation, i.e., recourse. The Actionable Recourse (ACTREC) method constrains the generated (exogenous) counterfactuals such that the alterations do not change immutable features. The problem is modeled through a mixed integer programming solved with CPLEX.[5] Differently to the previous formulation, in Ustun et al. (2019) $b(x+\delta) \neq b(x)$ is expressed in a constraint instead of in the cost function. The fact that valid solutions should be actionable is controlled through the set A of actionable features as constraints of the optimization problem, i.e., such that $\delta \subseteq A$. ACTREC is designed for tabular data and for differentiable classifiers. It can handle categorical features because all the numerical features must be discretized. However, the discretization can also be a limitation of this approach.

DACE. In Kanamori et al. (2020) the authors propose DACE, a Distribution-Aware Counterfactual Explanation method based on mixed integer linear optimization. The main novel contribution of DACE is the loss function that is based on the Mahalanobis distance and on the Local Outlier Factor (LOF) to evaluate the plausibility of candidate counterfactuals. The idea of DACE is to simultaneously minimize the distance and to keep also the counterfactual plausible with a low value of LOF. The CPLEX optimizer is used to solve the mixed integer linear optimization problem. DACE is designed to explain linear classifiers and tree ensembles. Depending on the model, DACE adopts a different set of constraints. Besides, DACE handles categorical features with one-hot encoding and recovers from their implausibility through the LOF score. It also accounts for actionability by allowing the definition of immutable features.

MACE. Karimi et al. (2020) propose MACE, a Model-Agnostic approach to generate Counterfactual Explanations. MACE is able to work on heterogeneous tabular data with any given distance function. MACE map the problem of counterfactual search into a sequence of satisfiability (SAT) problems. It expresses as logic formulae the black-box model, the distance function, the plausibility, actionability, and diversity constraints. The goal of each SAT problem is to check if exists a counterfactual at a distance smaller than a given threshold. Once that the nearest counterfactual is found, similarly to DCE, additional constraints can be inserted into the SAT problems to find alternative and diversity counterfactuals. MACE employs satisfiability modulo theories (SMT) solvers like Z3 or CVC4 to solve the SAT problems.

DICE. Diverse Counterfactual Explanations (DICE) Mothilal et al. (2020) solves an optimization problem with various constraints to ensure *feasibility* and *diversity* when returning counterfactuals. It returns a set of k plausible and different counterfactuals for the input x. The idea of DICE is to foster actionability and feasibility not only by allowing the user to specify the mutable and immutable features, but also through the diversity of the counterfactuals in C. DICE accounts for diversity (i) by extending the loss function to search for k counterfactuals, (ii) by adding a regularization term to the loss function that penalizes solutions formed by counterfactuals which are too similar:

[5] IBM CPLEX Optimizer https://www.ibm.com/analytics/cplex-optimizer.

$$\arg \min_{x'_1,\ldots,x'_k} \frac{1}{k} \sum_{i=1}^{k} \max(0, 1 - y' logit(b(x'_i))) + \frac{\lambda_1}{k} \sum_{i=1}^{k} d(x'_i, x) - \lambda_2 div(x'_1, \ldots, x'_k)$$

where *div* is the diversity metric that measure the sparsity among the counterfactuals, and λ_1, λ_2 hyper-parameters that balance the three parts of the loss function. DICE accounts for categorical features through one-hot encoding and adds a regularization term with high penalty for each categorical feature to force its values for different levels to sum to 1. DICE adopts the Adam optimizer. Mothilal et al. (2021) present an interesting work using WACH and DICE to generate features importance explanations from counterfactual explanations by labeling each feature as *necessary* or *sufficient*. They show that DICE and WACH do not agree with LIME and SHAP on the features importance ranking and that LIME and SHAP fail in identifying sufficient and necessary features.

C-CHVAE. C- CHAVE (Counterfactual Conditional Heterogeneous Autoencoder) is the model-agnostic explainer for tabular data presented by Pawelczyk et al. (2020). Also C- CHAVE makes use of an autoencoder that is used for modeling heterogeneous data and for approximating the conditional log-likelihood of the actionable attributes given the immutable ones. C- CHAVE does not require any distance function acting in the real input space. The CVAE adopted by C- CHAVE is used for both candidate counterfactual generation and to measure a regularization term in the loss function that, in this case, plays the role of the distance function as it determines the neighborhood of x in which C- CHAVE searches for counterfactuals. Finally, C- CHAVE can handle heterogeneous data types since the decoder is indeed a composition of various models, one per input, allowing to simultaneously model various data types.

SYNTH. Ramakrishnan et al. (2020) present SYNTH, a method for synthesizing action sequences for modifying model decisions. The idea of SYNTH is to find the least-cost, feasible sequence of actions, i.e., changes of feature values, such that $b(x') \neq b(x)$. SYNTH combines search-based program synthesis and optimization-based adversarial example generation to construct action sequences over a domain-specific set of actions. This combination enables SYNTH to handle differentiable black-box and categorical data. SYNTH is designed for tabular data but in Ramakrishnan et al. (2020) it is tested also on simple images. It starts from an empty sequence of actions. It picks an action sequence at every iteration and extends it with a new action from the set of possible actions. It solves an optimization problem with the Adam optimizer to find the new set of parameters. The search process continues until all sequences of some length are covered. Finally, it returns sequence with the minimal cost that changes the classification and satisfies all preconditions.

ARES. The Actionable REcourse Summaries approach (ARES) constructs *global* counterfactual explanations which provide an interpretable summary of recourses for an entire reference population Rawal and Lakkaraju (2020). ARES simultaneously optimizes for the validity of the counterfactuals and interpretability of the explanations, while minimizing the number of changes with respect to the reference population X. The optimization procedure is based on Lee et al. (2009). The initial rules can be provided by a user or extracted with Apriori. ARES also accounts for actionability

through a distance capturing the difficulty of changing a feature given the value of another one.

SCOUT. The Self-aware disCriminant cOUnterfactual explanaTion method of Wang and Vasconcelos (2020) (SCOUT), aims at returning *discriminant* counterfactual explanations for image classifiers. An explanation is produced by the computation of two *discriminant explanations* with the role of the input image x and an image with the desired class $\bar{x}$, inverted. A discriminant explanation for images x and $\bar{x}$ consists of a saliency map highlighting pixels highly informative for $b(x)$ but uninformative for $b(\bar{x})$. Discriminant explanations are obtained through an optimization process performed with an explanation architecture combining features activation layers of the CNN explained.

FRACE. Zhao (2020) presents FRACE (Fast ReAl-time Counterfactual Explanation), an explainer for neural networks classifiers for images. The architecture of FRACE is a neural network itself, and it is aimed at minimizing a loss function accounting for validity and a minimal perturbation. FRACE search for the perturbation through a starGAN (Choi et al. 2018) used as residual generator to generate the perturbation that causes the change of class. FRACE also accounts for plausibility because of the adversarial training. Experiments in Zhao (2020) show that it is markedly faster than SCOUT and CEM.

CEODT. Carreira-Perpiñán and Hada (2021) present CEODT, a Counterfactual Explanation method for Oblique Decision Trees. While most of the counterfactual explainers focus on differentiable black-box classifiers, CEODT is specifically designed for classification trees, and in particular for both traditional axis-aligned and oblique trees. Since for these models the counterfactual optimization problem is nonconvex, nonlinear, and nondifferentiable, CEODT computes an exact solution by the optimization problem within the region represented by each leaf, and then picking the leaf with the best solution. The problem solved has the form of a mixed integer optimization where the integer part is done by enumeration (over the leaves). This is possible for any type of tree. CEODT is able to work with high-dimensional feature vectors with both continuous and categorical features and also on different data types.

DECE. Cheng et al. (2021) present DECE, an interactive Decision Explorer with Counterfactual Explanation that provides explanations through a visualization system. The model behind the visualization system retrieves multiple exogenous counterfactuals by optimizing a loss function accounting for validity, distance minimality, number of changes, diversity. DECE allows specifying features constraints to account for actionability. The main innovation of DECE w.r.t. existing approaches is (i) the interactive framework, and (ii) the possibility to get insight when explaining subgroups of instances.

SGNCE. Mohammadi et al. (2021) present a counterfactual explanation approach specifically designed for neural networks based on optimization that provides guarantees for the minimality of the counterfactual returned as well as for the possibility to retrieve it. This second property is named *coverage* in the paper. Also, SGNCE allows to find a plausible and actionable counterfactual. The idea of SGNCE is to solve the

problem using mixed-integer programming. SGNCE can be seen as an evolution of MACE inheriting all its strengths.

OCEAN. Parmentier and Vidal (2021) present OCEAN, an Optimal Counterfactual ExplAiNer for tree ensembles. OCEAN employs an efficient mixed-integer programming approach to search for counterfactuals and Guroby to solve the mathematical models. The peculiarity of OCEAN is that the problem is formalized only through a number of binary variables, which is logarithmic in the number of vertices described by the decision trees forming the black-box model. From a certain perspective, it can be seen as an update of OAE and DACE but focused on tree ensembles. Besides efficiency, the strong points of OCEAN are that it accounts for both plausibility and actionability.

ORDCE. The Ordered Counterfactual Explanation proposed by Kanamori et al. (2021) accounts for asymmetric interaction among features, such as causality, by calculating a loss function that depends on the order of changing features. ORDCE not only returns the values of the features that must be updated $\delta_{x,x'}$ but extends them by returning the order in which they should be altered. Given a known interaction matrix among features, the order is considered into the mixed integer linear optimization approach through a penalization term in the loss function. Similarly to DACE, it can be used to explain different black-box models subject to the definition of appropriate constraints. It handles actionability and categorical features through one-hot encoding.

ALGREC. Karimi et al. (2021b) rely on causal reasoning to caution against the use of counterfactual explanations as a recommendable set of actions for recourse. Thus, ALGREC embeds a shift of paradigm from recourse via nearest counterfactual explanations. The idea is to find recourse through minimal interventions accounting for causality. Assuming a world where every feature is independent, then each valid counterfactual respecting actionability also respects causality. However, this is not the case in practice. Thus, ALGREC exploits a known causal model G capturing the dependencies among observed variables, and a family of actionable interventions A to solve an optimization problem that returns recourse through minimal interventions. ALGREC assumes that the causal model G falls in the class of additive noise models (ANM), so that it can compute the counterfactuals by performing the *Abduction-Action-Prediction* procedure proposed by Pearl et al. (2009).

PIECE. Kenny and Keane (2021) illustrate the PlausIble Exceptionality based Contrastive Explanation (PIECE) method for generating contrastive explanations for CNN working on image data. PIECE identifies feature-values with low probability in the latent features of the CNN representing the instance under analysis x, i.e., *exceptional features*, and attempts to modify them to be their expected values in the desired counterfactual class, i.e., *normal features*. We underline that the "features" treated by PIECE are not directly parts of the input image, but their latent features activating the neurons of the CNN. Finally, PIECE exploits a GAN to generate the counterfactual images.

CEGP. Van Looveren and Klaise (2021) propose CEGP, a method for Counterfactual Explanations Guided by Prototypes. CEGP adopts the same loss function employed by CEM. However, while CEM accounts for plausibility through a loss term calculated with an autoencoder, CEGP adopts also a loss term based on *prototypes* that guide the perturbation δ towards a counterfactual x' that respect the data distribution of class y'. CEGP defines a prototype for each class through the encoder of the autoencoder. The class prototype is defined as the average encoding over the k nearest instances

in the latent space with the same class label. Given an input feature x, CEGP first finds the nearest prototype in the latent space, and then use it to efficiently solve the optimization problem. CEGP accounts for categorical features by inferring distances between categories of a variable based on either model predictions, or on the context provided by the other variables. Then it applies multidimensional scaling to project the distances into one-dimensional Euclidean space, which allows CEGP to perform perturbations. Finally, it maps the resulting number back to the closest category to query the black-box. Like CEM it used the FISTA optimizer. Another important improvement of CEGP with respect to the other methods is that it is data-agnostic as it can be easily employed in any data type. In Balasubramanian et al. (2020) is proposed a similar approach but attempting to generate counterfactuals entirely in the latent space of a trained autoencoder or VAE. The same proposal already appeared in the explainers presented in Guidotti et al. (2019b) and in Guidotti et al. (2020).

POLYJUICE. POLYJUICE is a general-purpose counterfactual generator for textual data proposed by Wu et al. (2021). It returns a diverse set of realistic textual counterfactuals that can also be employed for explanation purposes. POLYJUICE accounts for similarity and minimality, diversity and plausibility in the sense that the generated counterfactual must be grammatically correct. This also guarantees endogenous counterfactuals. The counterfactual generation is performed as conditional text generation by fine-tuning the GPT-2 model Radford et al. (2019). The generation makes usage of a fill-in-the-blank structure to specify where the perturbation occurs and control codes like negation, delete, insert, shuffle, etc., to specify how it occurs.

4.4 Finding counterfactuals through heuristic search strategies

Another category of counterfactual explainers adopts Heuristic Search Strategies (HSS) to find/generate counterfactuals. Heuristic strategies are typically much more efficient than optimization algorithms. On the other hand, efficiency is paid with solutions that are not necessarily optimal. The search strategy is typically designed such that at each iteration, the solution x' is updated with the objective of minimizing a cost function. In turn, the cost function is based on a local and heuristic choice aimed at obtaining a valid counterfactual, which is also similar to x. In the following we describe the characteristics of counterfactual explainers based on heuristic search strategies.

SEDC. Described in Martens and Provost (2014), SEDC (Search for Explanations for Document Classification) is probably the first proposal for counterfactual explanation. It is a model-agnostic heuristic approach for textual data. The search is guided by local improvements via best-first search with pruning. SEDC starts by listing all potential explanations of one word obtained by removing from the instance under analysis x a single word and calculating the class and score change for each word. Then, SEDC proceeds with a best-first search. Given the current set of word combinations denoting partial explanations, it expands the partial explanation for which the output score changes the most in the direction of a class different from $b(x)$. Concerning the pruning, for each explanation with l words that, if removed, changes the prediction, SEDC do not check combinations of size $l + 1$ with these same words. Vermeire and Martens (2022) presents SEDCT, an extension of SEDC working on images and also

usable for multiclass classifiers. The method also takes as input an image segmentation function to binarize the image under analysis.

GIC. Lash et al. (2017b) propose GIC, a framework to solve the Generalized Inverse Classification problem. GIC recognizes non-actionable, directly actionable, and indirectly actionable features. Hence, it also handles causal relationships. The GIC framework offers three heuristic counterfactual explanation methods. The first one consists of a hill climbing plus local search procedure, the second one relies on a genetic algorithm, while the third one combines the genetic algorithm with the local search. All the heuristic methods exploit a bisection search to find the values for the counterfactual x'. GIC works on tabular data, does not handle categorical attributes nor validity due to heuristic procedures, and returns a single counterfactual. (Lash et al. 2017a) presents a first version of GIC acting only on differentiable models.

GSG. The Growing Spheres Generation (GSG) method illustrated by Laugel et al. (2018) relies on a generative approach growing a *sphere* of synthetic instances around x to find the closest counterfactual x'. Given x, GSG ignores in which direction the closest classification boundary might be. Indeed, GSG generates candidate counterfactuals Z randomly in all directions of the feature space until the decision boundary of b is crossed and the closest counterfactual to x is retrieved. GSG is a greedy approach. It starts by generating instances Z using a uniform distribution within a given radius η_0. The radius is halved until for all the instances $z \in Z$ we have that $b(z) = b(x)$. Then the previous radius is considered, and the most similar instance to z to x is returned as a valid counterfactual x'. We can say that GSG generates candidate counterfactuals in the feature space in a l2-spherical layer around x until a valid counterfactual is found. This gives the name to the algorithm.

POLARIS. It is a model agnostic explainer for neural networks that can be applied to any data type (Zhang et al. 2018). POLARIS adopt a heuristic search strategy for defining the regions of search to select the features to vary. Then the values are selected by solving an optimization problem solved through the Gurobi optimizer. POLARIS tries to guarantee stability by searching for counterfactual explanations which are valid not only for the instance under analysis, but also for those present in a region of interest with respect to a ε threshold parameter. The explanation returned is symbolic because it is formed by a set of rules describing a set of valid counterfactual instances.

CVE. The Counterfactual Visual Explanation (CVE) approach (Goyal et al. 2019) aims at finding counterfactual explanations for image classifiers by solving with greedy search approaches the minimum-edit counterfactual problem. The idea of CVE is simple but effective. Given a randomly selected image $\bar{x}$ with $b(\bar{x}) \neq b(x)$, CVE searches (with two possible strategies) for the minimum changes to x replacing with pixels selected from x', i.e., $x' = T(x, \bar{x})$, that leads to x' such that $b(x') = b(\bar{x})$ through a transformation T. CVE is model-agnostic but does not account for most of the properties in Table 1.

CADEX. Constrained ADversarial EXamples (CADEX), presented in Moore et al. (2019), is a method for generating counterfactual explanations for tabular data based on a heuristic that changes the differentiable model input with a minimal perturbation so that to obtain a different classification. The process is performed by minimizing the loss between $b(x)$ and $b(x')$ using an optimizer like Adam or RMSProp. CADEX accounts for sparsity by constraining the number of attribute changed and for plausibility by

constraining the direction of the gradient. Besides, CADEX also account for categorical data.

CFSHAP. In Rathi (2019) the authors presente an heuristic approach for counterfactual generation based on SHAP. In particular, given a record x, CFSHAP first estimates the Shapely values for each possible target class different from $b(x)$. Then, it randomly generates synthetic neighbors of x by permuting x only on the features for which the Shapely values are negative with respect to the desired counterfactual class. This approach has experimented only for tabular datasets formed by continuous attributes.

CERTIFAI. CERTIFAI is a model-agnostic and data-agnostic method for Counterfactual Explanations for Robustness, Transparency, Interpretability, and Fairness of Artificial Intelligence models Sharma et al. (2019). CERTIFAI exploits a genetic algorithm to generate counterfactuals. CERTIFAI considers a population of candidate counterfactuals where each candidate x' is a chromosome. The genetic procedure evolves the population through selection, mutation, and crossovers with the aim of maximizing a fitness function that accounts for the similarity between x' and x and constrained to be $b(x') \neq b(x)$. Finally, the population is restricted to the counterfactuals that respect the required constraints, and after a predefined number of iterations, those with the best fitness scores are returned in C.

PCATTGAN. In Arrieta and Ser (2020) the authors present a plausible counterfactual explainer relying on adversarial examples to retrieve counterfactuals. In particular, the PCATTGAN system comprehend an AttGAN model (He et al. 2019) and a multi-objective optimization model that infers the attribute modifications needed to produce plausible counterfactuals for the black-box b. The loss function accounts for validity, minimality and plausibility that is intended here as the implementation of credible changes not performed by a computer. As multi-objective optimizer is adopted the Speed-constrained Multi Objective Particle Swarm Optimization (SMPSO) (Nebro et al. 2009).

MOC. MOC is the Multi-Objective Counterfactuals explanation method presented in Dandl et al. (2020). The idea of MOC is to model the counterfactual search as a multi-objective *genetic* optimization problem such that the output C is formed by a set of diverse counterfactuals with different trade-offs between the required objectives. In particular, MOC accounts for four aspects modeled in the loss function: (i) the prediction outcome of $b(x')$ must be close to the desired output y', (ii) x' should be similar to the reference population X, (iii) x' differs from x only in few features, and (iv) it is an actionable and plausible instance. MOC is model-agnostic, works on numerical and categorical features on tabular data, and allows to specify actionable features to foster plausibility.

VICE. Gomez et al. (2020) present VICE, a method for Visual Counterfactual Explanations for machine learning models. The focus of the paper is on the visual part while the VICE algorithm implements a simple heuristic to find changes that are minimal in terms of both amount and number. VICE discretizes the dataset under analysis by fitting a Gaussian on each of the features and splitting the values into n bins and allows to select actionable features. It starts with the feature values of x and, in each iteration, it independently moves the value in each of the actionable features to the bins above and below the current one, and selects the one leading to the largest change in the

black-box prediction $b(x')$ in the direction of the desired outcome $y' \neq b(x)$. Then, it takes the maximum change across all the actionable features and uses it as starting point for the next iteration. Thus, VICE greedily moves feature values across the bins until the predicted class is changed, or until the constraints (no more than m' features can be changed) are violated.

PERMUTEATTACK. In Hashemi and Fathi (2020) the authors present PERMUTEATTACK, a model-agnostic counterfactual explainer for tabular data based on adversarial perturbation. PERMUTEATTACK solves the counterfactual explanation problem through a genetic algorithm that optimizes a fitness function accounting for validity and minimality in terms of both the number of changes and the distance in the Euclidean space. PERMUTEATTACK also deals with categorical attributes using both one-hot encoding or an out-of-distribution detection algorithm Carlini and Wagner (2017).

GRACON. Kang et al. (2020) present a method for counterfactual explanation based on GRAdual CONstruction for deep neural networks. The novelty of GRACON is that it accounts for the internal characteristics of the network to generate the counterfactual. Said in other words, GRACON tries to achieve plausibility with respect to the data manifold learned by the model to be explained. GRACON gradually constructs the counterfactual x' by iterating over masking and composition steps. Masking selects an important feature from x to obtain the class $b(x)$. This is achieved through the gradient of b. Composition optimizes the previously selected feature by ensuring that $b(x')$ is close to the logit space of the training data classified with the same label.

GRACE. Presented in Le et al. (2020), GRACE (GeneRAting Constrastive samplEs) is a counterfactual explainer designed to explain neural network working on tabular datasets. GRACE models the problem similarly to methods using optimization. However, it solves it through a heuristic contrastive sample generation algorithm that continuously perturbs x by projecting it on the decision boundary, and separating it with the desired (or nearest) class y'. To minimize the number of features varied, at each iteration GRACE only varies $m' < m$ features from an ordered list until it crosses the decision boundary. GRACE accounts for plausibility by projecting back the generated instances with a projection function that ensures that the final x' looks more real by checking feature range constraints. m' is gradually increased if the counterfactual candidates generated are not valid. The ordered list is obtained by prioritizing features that are highly affected by the outcome y'. Finally, GRACE generates an explanation in natural text using a template filled with the difference in features values available in $\delta_{x,x'}$. The difference in features values can be described as (i) extract value (e.g., 0.01 point lower), (ii) magnitude comparison (e.g., two times), or (iii) relative comparison (e.g., higher, lower).

MCBRP. In Lucic et al. (2020) the authors present the MCBRP method (Monte Carlo Bounds for Reasonable Predictions). The intuition behind MCBRP is to exploit Monte Carlo simulation as the heuristic search to identify unusual properties of a particular instance. Given an observation x, MCBRP generates as counterfactual explanation a set of upper and lower bounds for each feature that would result in a plausible and valid prediction with a different outcome. To help the user in understanding the explanation,

MCBRP also includes the relationship between each feature and the target, and how the input should be changed in order to change the output.

LIME-C/SHAP-C. In Ramon et al. (2020) the authors present a comparison between SEDC and LIME/SHAP re-adapted to return counterfactual explanations. The paper proposal is for textual data but given a function to binarize a dataset, LIME- C and SHAP- C can be applied to any data type. LIME- C and SHAP- C take as input x and runs LIME and SHAP, respectively, to extract the features importance ϕ. After that, SEDC is run on subsets of features with growing importance w.r.t. ϕ, i.e., first removing from x the most important feature, then removing from x the two most important features, etc., and stopping when a counterfactual is found. Results show that performance among the three methods are comparable in terms of validity and runtime. In Fernandez et al. (2020) is proposed an alternative similar to LIME- C/SHAP- C.

CLEAR. White and d'Avila Garcez (2020) propose CLEAR a method for Counterfactual Local Explanations via Regression. CLEAR provides counterfactuals that are explained by regression coefficients, including interaction terms. First, CLEAR generates a random synthetic neighborhood around x and selects a small balanced sub-sample composed of instances at diversified level of probability $b(x')$ from $b(x)$ according to predefined parameters modeling the margins around the decision boundary. Then for each feature, finds a counterfactual instance varying only a feature with a brute force approach and extends the balanced neighborhood with them. Finally, it trains a local surrogate linear regressor r on the balanced neighborhood and estimates the counterfactual instances retrieved at the previous step. CLEAR returns as explanations the actual and estimated counterfactuals as well as the regressors unveiling the feature coefficients and the approximation error between b and r. Thus, as a heuristic search strategy, CLEAR adopts a post-hoc procedure similar to the one used by LIME (Ribeiro et al. 2016). CLEAR is model-agnostic, handles categorical features, and returns more than one exogenous counterfactual.

PCIG. PCIG is a method for Plausible Counterfactual Instances Generation for textual data presented by Yang et al. (2020). A counterfactual explanation in the context of a text is composed of a word (or by a set of words) that, if inserted or substituted in a plausible way in the input sentence, then the outcome of the prediction is changed. PCIG generates grammatically plausible counterfactuals by replacing the most important words with the antonyms based on pre-trained language models. Thus, after having learned the word importance, PCIG identifies the words responsible for flipping the outcome by replacing them with the intersection of grammatically plausible substitutes using masked language model and words in the reverse emotional dictionary.

GECO. The GEnetic COunterfactual explainer (GECO) is presented in Schleich et al. (2021). GECO retrieves diverse counterfactuals through a genetic algorithm and automatically accounts for plausibility and accountability during the mutation and crossover operations thanks to the usage of PLAF constraints. PLAF is a plausibility-feasibility constraint language defined in the paper. It is designed for tabular data and also handles categorical attributes. It offers a library in Julia that addresses inefficiencies of many counterfactual explainers by grouping similar records w.r.t. specific subset of attributes, and partial evaluation of the black box classifier.

4.5 Finding counterfactuals with instance-based strategies

The very simple but effective idea of instance-based (or case-based) approaches for counterfactual explanation is to search into a reference population instances to be used as counterfactuals.

NNCE. The Nearest-Neighbor Counterfactual Explainer (NNCE) is an endogenous counterfactual explainer inspired by NN classifiers (Shakhnarovich et al. 2008) that select as counterfactual(s) the instances in $x' \in X$ most similar to x and with a different label, i.e., $b(x') \neq b(x)$. Candidate counterfactuals are sorted with respect to the similarity with x and the k most similar ones are selected as a result and placed in C. A weakness of NNCE is the computational cost of computing distances between x and all the instances in X with a different outcome. This negative point can be recovered by using a sample of X to search for a counterfactual instance. This reduces the complexity but increases the probability of having a counterfactual substantially different from x. Another weakness of this approach is that it only accounts for similarity and validity, but not for diversity (even though more than a counterfactual can be returned) and minimization of changes. NNCE can be empowered to account for actionability by (i) calculating the distances only over the space of actionable features, and (ii) by ensuring that the immutable features are not modified by overwriting non-actionable features of x' with their value in x. This update makes it necessary for NNCE to check if the candidate counterfactuals are still valid after the feature overwriting. In addition, accounting for actionability in this simple way can turn NNCE into an exogenous explainer. The What-If tool[6] presented in Wexler et al. (2020) is a visual instrument offering a way to envision counterfactual explanations on bi-dimensional plots for small datasets. The counterfactual selection is performed through the NNCE approach using the L1 or L2 distance functions.

CBCE. Keane and Smyth (2020) present CBCE, a Case-Based Counterfactual Explainer realized as a refinement of NNCE. CBCE adopts the notion of *explanation case* that, given a reference dataset X, consists in couples of instances (x, x') such that (x, x') are the two most similar instances in X and $b(x) \neq b(x')$ holds. Then, given an input instance p, CBCE first identifies among the available explanation cases the couple (x, x') having the most similar instance with the same outcome x, i.e., $b(x) = b(p)$. After that it creates a candidate counterfactual p' initializing it with the values of p and by replacing the different features of x' between x and x' in p'. Hence, p' is a combination of feature values from p and x'. The idea is that p' differs from p in a similar way to the manner in which x' differs from x. Finally, if $b(p') = b(x')$ a valid counterfactual is found, otherwise another explanation cases can be used for an additional adaptation step. Since CBCE generates counterfactuals by copying values from existing instances, it is defined as an endogenous explainer. CBCE is not designed to handle categorical features, but it is just a matter of the distance function adopted. Moreover, it can be employed on any data type even though it is experimented only on tabular datasets.

FACE. Feasible and Actionable Counterfactual Explanations (FACE) Poyiadzi et al. (2020) focuses on returning "actionable" counterfactuals by uncovering "feasible

[6] https://pair-code.github.io/what-if-tool/explore/.

paths" for generating counterfactual. These feasible paths are the shortest path distances defined via density-weighted metrics. In this way, FACE extracts plausible counterfactuals that are coherent with the input data distribution. More it details FACE works as follows. First, it generates a graph over the data points by using KDE, k-NN or an ϵ-graph. The user can also select the prediction threshold of b, the density threshold, the weights of the features, and custom condition functions to specify actionability. Then, it updates the graph according to these constraints. Finally, FACE applies a shortest path algorithm to find all the data points that satisfy the requirements. FACE is an endogenous and data-agnostic counterfactual explanation method that can theoretically be used also to work on datasets with categorical features.

NICE. Brughmans and Martens (2021) present NICE, an algorithm for nearest instance counterfactual explanations. NICE is model-agnostic, works on tabular data, and deals with categorical features. The authors propose four versions of NICE. The base version corresponds with NNCE with training samples correctly classified and using as distance the heterogeneous Euclidean overlap method (Wilson and Martinez 1997). The other three versions start from x and iteratively permute one feature after the other with the objective of turning x into x' where x' is the closest counterfactual to x returned by NNCE. Finally, the three versions of NICE return the counterfactuals that maximize a reward function accounting for sparsity, diversity, or plausibility, respectively. Similarly to CBCE, NICE can be considered an endogenous explainer.

4.6 Finding counterfactuals exploiting decision trees

Decision trees (DT) are a simple model to identify and/or generate counterfactual explanations. The main idea is to use a decision tree to approximate the behavior of the black-box classifier and then exploit the logic revealed by the tree for building the counterfactual explanations. This idea comes from post-hoc explanation methods such as Craven et al. (1995) that highlights how decision tree, due to their structure, are appropriate surrogate models to unveil the logic of decision systems. Indeed with decision trees it is easily possible to reason into counterfactual terms by moving along the tree structure. On the other hand, other simple surrogate models, such as linear models, are not appropriate for counterfactual reasoning because they are only composed of coefficients that do not allow logical reasoning. In the following, we describe how decision trees can be employed to find the counterfactual examples.

TBCE. A Tree-Based Counterfactual Explainer (TBCE) exploits a surrogate (shadow) decision tree $\mathcal{T}$ trained on a *reference dataset* X to mime the behavior of the classifier b. Depending on X, the tree T can capture a different logic, and therefore the counterfactuals returned can highlight different attributes to be changed. First, X can be the training set of the black-box b, it can be sampled from the training set, or it can be a synthetic dataset. Second, X can be "global" modeling the whole data distribution, or "local", modeling only the data around the instance under analysis (Guidotti et al. 2019c). Leaves in the decision tree T leading to predictions different from $b(x)$ can be exploited for building counterfactuals. Basically, the splits on the path from the root to one such leaf represent conditions satisfied by counterfactuals such that $b(x) \neq y'$. Actionability can be ensured by considering only splits involving action-

able constraints. To tackle minimality, the counterfactual paths are sorted with respect to the number of conditions not already satisfied by x. Then for each such path, TBCE chooses one instance x' from X, reaching the leaf and minimizing the distance to x. If TBCE stops at this point, then the counterfactuals are endogenous with respect to X as they are selected from X itself. However, even though the path has been checked for actionable splits, the candidates x' may still include changes with respect to x that are not actionable. This weakness can be overcome by overwriting non-actionable features with the values of x. As a consequence of the aforementioned correction, and due to the fact that not all the candidates are necessarily valid, TBCE has to check the validity of x' controlling that $b(x') \neq b(x)$ before including in the result set C. The search over different paths of the decision tree allows for some diversity in the results, even though this cannot be explicitly controlled.

FT. Tolomei et al. (2017) present a method based on actionable Feature Tweaking (FT) to understand which adjustable features of a given instance x should be modified to alter the prediction of a tree-based ensemble. FT is designed to explain tree-based ensemble trained for binary classification on tabular data. In particular, it exploits the internals of the ensemble to retrieve the recommendations for transforming instances from a class to another one, i.e., $\delta_{x,x'}$, with respect to a set of given actionable features. The main idea is the same illustrated for TBCE with the substantial difference that this time the validity must be respected for all the trees in the ensemble and not just one. Indeed, given a candidate counterfactual x' we could have that for a specific tree in the ensemble we have that $T_i(x') = y'$ while for the ensemble $b(x')$ is still equal to y, i.e., the alterations which are affecting the i-th are not affecting the majority of the trees in the ensemble. The goal of FT is to find x' such that for the majority of the trees $b(x') = y'$. This task is achieved by considering a positive threshold that bounds the tweaking of every single feature to pass every boolean test on a positive path of each tree. FT handles with categorical features, returns more than a counterfactual, and the counterfactual are endogenous because they are obtained from the tweaking among existing values as ensemble classifiers cover the whole feature domain.

LORE. The LOcal Rule-based Explainer (LORE) (Guidotti et al. 2019a), is a local agnostic method that provides explanations in the form of rules and counterfactual rules. Given a black-box b and an instance x, with $b(x) = y$, LORE first generates a set of synthetic neighbors Z through a genetic algorithm such that for some instances $b(z) = b(x)$ and for some other instances $b(z) \neq b(x)$. Then, it trains a decision tree T on this set labeled with the black-box outcome $b(Z)$ and from the tree retrieves the *factual* decision rule, that corresponds to the path on the decision tree followed by the instance x to reach the decision y, and *(ii)* a set of counterfactual rules, which have a different classification w.r.t. y. This counterfactual rules set shows the conditions that can be varied on x in order to change the output decision. LORE is explicitly tailored for tabular data. However, in Guidotti et al. (2019b, 2020), Lampridis et al. (2020) is shown how LORE can be extended to work on image data, time series, and textual data through the usage of autoencoders. The conditions in the counterfactual rules can be paired with the feature values for changing the outcome listed in $\delta_{x,x'}$. Through the

surrogate tree and the local neighborhood (that is used by LORE as reference set), with LORE it is possible to select counterfactual examples on the leaves of the tree.

FOILTREE. In Van Der Waa et al. (2019) the authors present a method returning contrastive explanations with local foil trees. FOILTREE works similarly to LORE by training a local surrogate tree in the neighborhood of x. The neighborhood is obtained either randomly sampling from an existing dataset, or generated according to normal distributions like LIME. Closeness is obtained by weighting the instances w.r.t. their distance to x. The explanation is build as difference between the nodes of the tree leading to the fact-leaf with those leading to the foil, i.e., contrastive, one. Like for LORE, counterfactual instances can be selected for the records respecting the contrastive rule.

RF-OCSE. In Fernández et al. (2020) the authors present the Random Forest Optimal Counterfactual Set Extractor (RF- OCSE), a method to extract counterfactual sets from a Random Forest (RF). In this setting the term *counterfactual sets* is used to indicate counterfactual rules, i.e., sub-region of the feature space where the counterfactual $b(x') \neq b(x)$ holds. As already discussed for FT, counterfactual explanation extraction for a tree-ensemble differs from a single decision tree because a consensus among the individual tree predictors is needed. RF- OCSE overcomes this limitation by converting a RF into a single DT, and then it extracts the counterfactuals from the tree similarly to TBCE. In the literature, there are various techniques to merge a set of trees into a single decision tree (Fan and Li 2020; Strecht 2015).

4.7 Other counterfactuals explanation methods

We briefly report here other explainers that do not fit in the taxonomy above because they are not explicitly meant to return counterfactual explanations but can be adapted to this task, because they are not applied to explain classifiers, or because they act on data types not studied above such as graphs.

ANCHOR. The explanation method ANCHOR presented in Ribeiro et al. (2018), even though it is not a counterfactual explainer, it is worth mentioning because its objective is to retrieve the explanations in the form of "sufficient" conditions for the classification. Hence, to some extent, it can be considered as the opposite of counterfactual explanations.

PRINCE. PRINCE is a counterfactual explainer for recommender systems presented in Ghazimatin et al. (2020) generating explanations as a minimal set of actions on a heterogeneous information network, i.e., graphs, used to describe the possible user interactions.

SURV-CF. Kovalev et al. (2021) propose a method for counterfactual explanation of machine learning survival models. The difficulties of counterfactual explanation in this setting is that the classes of examples are defined through form of survival functions. The authors introduce a condition that establishes the difference between survival functions and the counterfactuals. The problem is reduced to a standard con-

vex optimization problem with linear constraints and solved through Particle Swarm Optimization algorithm.

COMTE. In Ates et al. (2021) the authors present COMTE, a counterfactual explanation methods for multivariate time series. COMTE adopts an idea similar to CVE as it starts the counterfactual generation from a "distractor" instance $\bar{x}$ classified with the desired counterfactual class. Then changes x by randomly selecting subparts of $\bar{x}$ through a greedy procedure implemented with a random-restart hill climbing approach that aims at miniminzing a loss function accounting for validity and similarity.

CF-GNNExplainer. Lucic et al. (2021) propose CF- GNNEXPLAINER, a a counterfactual explainer for classifiers working on graphs. The counterfactual is formed by the minimal perturbation to the graph such that the prediction changes. Perturbations are expressed in the form of edge deletion.

MEG. In Numeroso and Bacciu (2021) the authors propose MEG (Molecular Explanation Generator). MEG generate counterfactual explanations for graph neural networks under the form of valid compounds with high structural similarity and different predicted properties. MEG trains a reinforcement learning generator but restricts the action space of the in order to only keep actions that maintain the molecule in a valid state.

5 Evaluation metrics

In line with Guidotti (2021), Visani et al. (2020) we believe that is important to formalize a set of evaluation measures to estimate the goodness of counterfactual explanations and counterfactual explainers. Therefore, we evaluate the performances of counterfactual explainers with respect to the various properties highlighted in Sects. 3.1 and 3.2. The measures reported in the following are stated for a single instance x to be explained, and considering $C = f_k(x, b, X)$ the returned counterfactual set. The metrics are obtained as the mean value of the measures over all x's to explain.

Size It measures the number of available counterfactuals. Indeed, the number of counterfactuals $|C|$ can be lower than k. We define $size = |C|/k$. The higher the better. Recall that by definition of a counterfactual explainer, $x' \in C$ are valid, i.e., $b(x') \neq b(x)$. Our intent is not to penalize methods designed to return a single counterfactual. Indeed, we account for the size only when comparing explainers returning more than a counterfactual. In a similar fashion, the *coverage* proposed by Mohammadi et al. (2021) measures if a counterfactual explainer can return explanations for all the instances analyzed.

Dissimilarity It measures the proximity between x and the counterfactuals in C. The lower the better. We measure it in two fashions. The first one, named dis_{dist}, accounts for similarity or proximity, and it is the average distance between x and the counterfactuals in C where the usage of different distance functions d can return different results. The second one, dis_{count}, accounts for minimality or sparsity, and it quantifies the average number of features changed between a counterfactual x' and x. The $\mathbb{1}_{cond}$ operator returns 1 if *cond* is true, and 0 otherwise. Let m be the number of features.

$$dis_{dist} = \frac{1}{|C|} \sum_{x' \in C} d(x, x') \qquad dis_{count} = \frac{1}{|C|m} \sum_{x' \in C} \sum_{i=1}^{m} \mathbb{1}_{x'_i \neq x_i}$$

Implausibility It measures the level of plausibility of the counterfactuals C. It accounts for how close are counterfactuals to the reference population X. It is the average distance of $x' \in C$ from the closest instance in the known set X. The lower the better.

$$impl = \frac{1}{|C|} \sum_{x' \in C} \min_{x \in X} d(x', x)$$

We underline that, this way we adopted to evaluate implausibility does not necessarily account for combinations of feature-values forming a non plausible record. Indeed, the measure adopted here is just a proposal among different alternatives that could be used to estimate the level of implausibility, e.g., outlier detection approaches like the Local Outlier Factor or Isolation Forests (Breunig et al. 2000; Guidotti and Monreale 2020; Liu et al. 2008).

Discriminative Power It measures the ability to distinguish through a naive approach between two different classes only using the counterfactuals in C. In line with Kim et al. (2016), Mothilal et al. (2020), we implement it as follows. We define the sets $X_{=} \subset X$ and $X_{\neq} \subset X$ such that $b(X_{=}) = b(x)$ and $b(X_{\neq}) \neq b(x)$ by selecting the instances in $X_{=}$, $X_{\neq}$ which are the k closest to x. Then, we train a simple 1-Nearest Neighbor (1NN) classifier using $C \cup \{x\}$ as training set, and d as a distance function. The choice of 1NN is due to its simplicity and connection to human decision making starting from examples. Finally, we classify the instances in $X_{=} \cup X_{\neq}$ and we use the accuracy of the 1NN as *discriminative power* (*dipo*). The higher the better.

Actionability It measures the level of actionability of the counterfactuals C and accounts for the counterfactuals in C that can be realized. Given the set A of actionable features, we measure the accountability as $act = |\{x' \in C \mid a_A(x', x)\}|/k$, where the function $a_A(x')$ returns true if x' is actionable w.r.t. the list of actionable features A, false otherwise. The higher the better.

Diversity It accounts for a diverse set of counterfactuals, where different actions can be taken to change the decision. The higher the better. We denote by div_{dist} the average distance between the counterfactuals in C, and by div_{count} the average number of different features between the counterfactuals.

$$div_{dist} = \frac{1}{|C|^2} \sum_{x' \in C} \sum_{x'' \in C} d(x', x'') \quad div_{count} = \frac{1}{|C|^2 m} \sum_{x' \in C} \sum_{x'' \in C} \sum_{i=1}^{m} \mathbb{1}_{x'_i \neq x''_i}$$

Instability It measures to which extent the counterfactuals C obtained for x are close to the counterfactuals $\bar{C}$ obtained for $\bar{x} \in X$, where $\bar{x}$ is the closest instance to x and $\bar{x}$ receives the same black-box decision of x, i.e., $b(x) = b(\bar{x})$. The rationale is that similar instances x and $\bar{x}$ should obtain similar explanations (Guidotti and Ruggieri

2019). The lower the better.

$$inst_{x,\bar{x}} = \frac{1}{1 + d(x, \bar{x})} \frac{1}{|C||\bar{C}|} \sum_{x' \in C} \sum_{x'' \in \bar{C}} d(x', x'')$$

with $\bar{x} = argmin_{x_1 \in X \setminus \{x\}, b(x_1) = b(x)} d(x, x_1)$ and $\bar{C} = f_k(\bar{x}, b, X)$. Such a measure can be used also to evaluate the instability for the same instance in case of multiple runs, i.e., with $\bar{x} = x$. We highlight that from another perspective of stability, in Slack et al. (2021) is shown counterfactual explanations may converge to different counterfactuals under a small perturbation of the reference population X, indicating that they are not robust.

Runtime It measures the efficiency in terms of elapsed time required by the explainer to compute the counterfactuals. The lower the better.[7]

In line with Mothilal et al. (2020), Wachter et al. (2017), in the above evaluation measures we adopt as distance d a mixed distance defined as:

$$d(a, b) = \frac{1}{m_{con}} \sum_{i \in con} \frac{|a_i - b_i|}{MAD_i} + \frac{1}{m_{cat}} \sum_{i \in cat} \mathbb{1}_{a_i \neq b_i}$$

where *con* (resp., *cat*) is the set of continuous (resp., categorical) features.[8] It is worth mentioning that in Mahajan et al. (2019), Mothilal et al. (2020) the evaluation measures employing a distance function are reported separately for continuous and categorical features.

We do not define metrics for causality because, to the best of our knowledge, there are no papers in the literature that have defined a way to measure to which extent causality is respected. Also, we highlight that these metrics are selected and reported because they are already used in other published papers and can contribute to assessing the various desirable properties for counterfactuals. For instance, there is a hidden trade-off between *size* and dissimilarity. The duality of these measures makes unclear if many counterfactuals with a single change are better than fewer counterfactuals with more changes.

[7] We underline that, in this setting, given a certain value k of counterfactuals required by a user, we do not consider if a method has incremental running time for higher k or another one requires the same time for $k = 1$ or $k = 100$. What matters is just the requirement of the user. Then the faster a method is the better it is for the user.

[8] We highlight that we experimented with several other distance functions obtained as a combination of the following: Euclidean, Cosine, Mean Absolute Deviation for continuous features, Mismatch, Jaccard, or Hamming for categorical features, or by using the Euclidean distance for all the features. However, the results obtained were comparable, therefore, in order to be coherent with the literature and for brevity, we report only those obtained with the same distance function adopted in Mothilal et al. (2020), Wachter et al. (2017).

6 Experiments

In this section, we benchmark a set of counterfactual explainers that offer an easily usable and well-documented library.[9] Also, we decided to implement a set of counterfactual explainers simple to realize to compare at least a method for each one of the categories analyzed in the taxonomy. We maintain separated the experimentation between explainers of existing libraries and the custom ones. After presenting the experimental setting, (i) we report a counterfactual explainers demonstration pointing the attention on actionability, and (ii) we illustrate a large quantitative validation comparing state-of-the-art explainers.

6.1 Experimental setting

We experimented on various datasets largely adopted as a benchmark in the literature. In particular, since most of the explainers in the literature works on tabular data, we focus our evaluation on four tabular datasets[10] described in Table 2 for which the instances describe attributes of an individual person, and the decisions taken by a black-box target socially sensitive tasks such as income estimation, loan acceptance, risk of recidivism, etc. For every dataset, we identify with n the number of instances, m the number of features, m_{con} and m_{cat} the number of continuous and categorical features, respectively. When necessary, accordingly to the black-box and to the explainer, for categorical features we adopt a one-hot binary encoding passing in this way from m features to m_{1h} features. For every dataset, we recognized a set of m_{act} features which are actionable, or not actionable, i.e., the counterfactual cannot contain a variation of the values for these features. We selected the following sets of not actionable features[11] `adult`: age, education, marital status, relationship, race, sex, native country; `compas`: age, sex, race; `fico`: external risk estimate; `german`: age, people under maintenance, credit history, purpose, sex, housing, foreign worker. Also, with the exception of `fico`, all the datasets have both continuous and categorical features, which requires a one-hot encoding step.

For every dataset, we trained and explained the following black-box classifiers: Random Forest (RF) as implemented by *scikit-learn*, and Deep Neural Networks (DNN) implemented by *keras*. We split the datasets into a 70% partition used for the training and 30% used for the test using a stratified partitioning with respect to the target variable. For each black-box and for each dataset, we performed on the

[9] The Python code, the datasets, and the scripts for reproducing the experiments are publicly available at https://github.com/riccotti/Scamander. Experiments were performed on Ubuntu 20.04 LTS, 252 GB RAM, 3.30 GHz × 36 Intel Core i9. With "easily usable library" we referred to well-documented libraries offering interfaces/functions not requiring an excessive training time to be used and understood and returning outputs formatted in similar and simple ways. We also implemented from scratch the most simple explainers.

[10] https://archive.ics.uci.edu/ml/index.php, https://www.kaggle.com/datasets.

[11] Regarding the selection of actionable features, we followed common sense mainly driven by the semantic of the names of the attributes. We underline that the main idea is to constrain certain features with respect to others and check how different methods behave.

Table 2 Datasets description and black-box accuracy: n number of records, m number of features, m_{con} number of continuous features, m_{cat} number of categorical features, m_{act} number of actionable features, m_{1h} number of features after one-hot encoding, l number of labels (classes)

Dataset	n	m	m_{con}	m_{cat}	m_{act}	m_{1h}	l	RF	NN
`adult`	32,561	12	4	8	5	103	2	.85	.84
`compas`	7214	10	7	3	7	17	3	.56	.61
`fico`	10,459	23	23	0	22	–	2	.68	.67
`german`	1000	20	7	13	13	61	2	.76	.81

training set a random search with a five cross-validation for finding the best parameter setting.[12] The classification accuracy on the test sets is shown in Table 2 (right).

We benchmark the following counterfactual explainers. DICE offers an implementation that handles categorical features, actionability, and allows to specify the counterfactuals k to return but is not model-agnostic as it only explains differentiable models such as DNNs. The *FAT* (Sokol et al. 2019) library offers a brute force (BF) counterfactual approach that handles categorical data and actionable features but does not allow to specify how many counterfactuals k must be returned. However, the *FAT*-BF library can return more than a counterfactual also if the user is not able to specify the required number k. Indeed, in the following, even though k is varied, the metrics returned for *FAT*-BF are fixed, resulting in a straight line. The *ALIBI* library offers the explainers CEM, CEGP and WACH. All of them are designed to explain DNNs, do not handle categorical features, and return a unique counterfactual, but it is somehow possible to handle actionability by playing with the admissible feature ranges. Even though CEM, CEGP and WACH are *theoretically* designed to explain only DNNs but, passing the prediction function instead of the model, they still work because their implementation practically rely only on prediction probability function. Thus, we experimented with them also on the RF of *sklearn*. We also experimented with CEML (Artelt 2019), a model-agnostic toolbox for computing counterfactuals based on optimization that does not handle categorical features but handles actionability. For the explanation methods not handling categorical features, i.e., WACH, CEM, CEGP and CEML, we adopted a one-hot encoding representation. In addition, we implemented the case-based counterfactual explainer (CBCE) to experiment with an endogenous explainer based on distances. For each tool, we adopt the default parameter setting offered by the library or suggested in the reference paper. We measure the actionability for both methods guaranteeing it by design and also for those not considering it. It is important to check correctness of the methods, but it is also a way to check to which extent the counterfactuals returned own this desired property without being controlled on this aspect. We did not experiment with DECE as it only works with PyTorch models.

In addition, we implemented the following counterfactual explainers which are not based on optimization strategies: GSG that follows the heuristic search based on growing spheres; NNCE that selects counterfactuals according to the nearest neighbor

[12] Details of the parameters can be found in the repository.

principle[13]; TBCE that approximate the black-box with a decision tree and retrieves counterfactuals from the tree structure; RCE that implements a random counterfactual explanator that randomly changes the values of the features until k valid counterfactuals are found; a re-implementation of the brute force counterfactual search procedure BF that allows specifying the maximum number m' of features that can be varied and the number of values to test for each feature r. The following variants are considered: NNCE- S runs the nearest neighbor search only on a random sub-sampling of 100 instances of the reference population to speed up the calculus; TBCE- P is TBCE with a decision tree pruned at a maximum depth of 4; BF1 and BF2 are BF with max number of features 1 and 2, respectively. For all these approaches requiring a distance and acting on datasets with continuous and categorical attributes, we adopted by default a mixed distance weighting Euclidean distance for continuous features and the Jaccard dissimilarity for categorical features. We show in the experiments how a different distance function can affect their results.

We do not benchmark against well-known explainers such as LIME or SHAP because the type of explanation returned is intrinsically different, and we believe that comparing against them is out of the purpose of this survey. Also, methods like LIME- C, SHAP- C and CFSHAP do not offer library sufficiently customizable for our experiments.

6.2 Counterfactual explainers demonstration

We report here a demonstration comparing the counterfactuals returned by the methods analyzed, focusing on actionability, diversity, and size of the different explanations. Figure 1 shows the counterfactual examples returned when explaining the DNN on the `german` dataset for an instance[14] x classified as *loan denied* with $k = 3$. For each counterfactual, only the feature values different from x are shown, i.e., the $\delta_{x,x'}$. In Fig. 1 we short the notation simply writing x' to intend $\delta_{x,x'}$. CEM, CEGP and WACH were not able to find any counterfactual (we tried different parameters). CEML only returns one counterfactual, hence not respecting the parameter $k = 3$. Also NNCE returns only two counterfactuals to respect the actionability. DICE and TBCE return counterfactuals with many changed features. Some of the features changed by DICE (highlighted in red) are not actionable. The counterfactuals of CBCE are not minimal and not quite similar among each other. The counterfactuals returned by BF are actionable and minimal but do not foster diversity, as there is one counterfactual repeated twice. Perhaps through this visual inspection, the "best" counterfactuals could be judged those of GSG as they are different, quite minimal, and actionable. However, from the following experiment will emerge that GSG is not too stable, and therefore, a re-run for the same instance could lead to alternative and different solutions.

[13] We cannot employ What-If (Wexler et al. 2020) as it only offers a visual interface.

[14] We report a demonstration only for one dataset/black-box because of space limitations.

$x =$ {*duration_in_month = 20, credit_amount = 3271, installment_as_income_perc = 2.97, present_res_since = 2.84, age = 24, credits_this_bank = 1.40, people_under_maintenance = 1, account_check_status = "0 ≤ ⋯ < 200 DM", credit_history = "existing credits paid back duly till now", purpose = "domestic appliances", savings = "100 ≤ ⋯ < 500", present_emp_since = "< 1 year", personal_status_sex = "male : married/widowed", other_debtors = "none", property = "if not A121/A122 : car or other, not in attribute 6", other_installment_plans = "none", housing = "rent", job = "skilled employee", telephone = "yes", "foreign_worker = yes"* }

$b(x) =$ *denied loan*

DICE

$x'_1 =$ { *duration_in_month = 23.91, credit_amount = 4964, installment_as_income_perc = 4.09, present_res_since = 3.39, age = 35.54* }

$x'_2 =$ { *duration_in_month = 29.94, credit_amount = 5528, installment_as_income_perc = 4.09, present_res_since = 3.39, age = 35.54, credits_this_bank = 1.84, credit_history = "all credits at this bank paid back duly", purpose = "(vacation - does not exist?)", personal_status_sex = "female : divorced/separated/married", other_debtors = "co-applicant", other_installment_plans = "bank", housing = "for free"* }

$x'_3 =$ { *duration_in_month = 29.94, credit_amount = 3835, installment_as_income_perc = 4.09, present_res_since = 3.39, age = 35.54, credits_this_bank = 1.84* }

BF

$x'_1 =$ { *telephone = "none"* }

$x'_2 =$ { *telephone = "none"* }

$x'_3 =$ { *other_debtors = "co-applicant"* }

CEML

$x'_1 =$ { *telephone = "none"* }

CBCE

$x'_1 =$ { *installment_as_income_perc = 1.85, present_res_since = 1.74, savings = "⋯ < 100 DM", property = "real estate", telephone = "none",* }

$x'_2 =$ { *savings = "⋯ < 100 DM", present_emp_since = 1 <, telephone = none* }

$x'_3 =$ { *account_check_status = "⋯ ≥ 1000 DM", savings = "⋯ < 100 DM", present_emp_since = "1 ≤ ⋯ < 4 years", other_debtors = "co-applicant", property = "real estate", other_installment_plans = "stores", telephone = "none"* }

GSG

$x'_1 =$ { *other_debtors = "co-applicant"* }

$x'_2 =$ { *other_debtors = "co-applicant", present_emp_since = "unemployed",* }

$x'_3 =$ { *telephone = "none"* }

NNCE

$x'_1 =$ { *duration_in_month = 32.95* }

$x'_2 =$ { *duration_in_month = 57.06* }

TBCE

$x'_1 =$ { *credits_this_bank = 1.98, property = "unknown / no property" other_installment_plans = "stores" job = "management/ self-employed/ highly qualified employee/ officer" telephone = "none"* }

$x'_2 =$ { *credits_this_bank = 1.98, property = "unknown / no property" other_installment_plans = "bank" job = "unskilled - resident" telephone = "none"* }

$x'_3 =$ { *savings = "⋯ < 100 DM", present_emp_since = 1 <, telephone = "none"* }

Fig. 1 Explanations for an instance x of the `german` classified as a *denied loan* by a DNN. In red, non-actionable changes (Color figure online)

6.3 Quantitative evaluation

In the following, in line with Mothilal et al. (2020), we report the performance of the selected counterfactual explainers varying the required number of counterfactuals k. The explainers that return a single counterfactual have constant values on each plot. We report also them in the plots in order to compare all the methods simultaneously. For every dataset and black-box, we explain 100 instances of the test set (not observed by the explanation method). We report aggregated results as means among the various instances, datasets, and black-box classifiers. Indeed, the minor deviations among

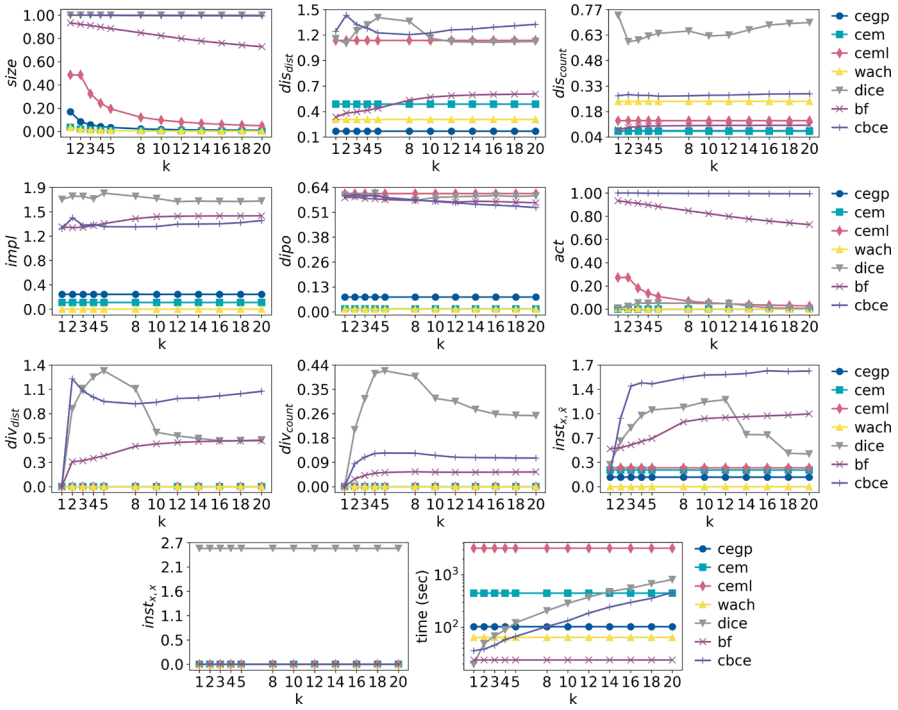

Fig. 2 Aggregated metrics for explainers implemented by existing libraries varying the required number of counterfactuals k. Best view in color (Color figure online)

different datasets and black-box do not justify the need to separate the results, also because of space limitations. In the following, every plot shows how an evaluation measure (y-axes) varies when changing the number of counterfactuals k (x-axes). Each line represents a different method with a unique color and marker shape.

Figure 2 shows the performance of counterfactual explainers implemented by existing libraries. From the first plot (*size*), we notice that, among the explainers able to return more than a counterfactual only DICE, CBCE and BF are able to return at least 80% of the required counterfactuals. We highlight that the other methods in many cases are not even able to return a single counterfactual as the reported average value of *size* is lower than one even for $k = 1$. This aspect highlights that methods that look for more than a solution have more chance of not returning an empty set.

Concerning *dissimilarity*, CEGP and WACH are the best performer when observing dis_{dist}, while CEM and BF when observing dis_{count}. This highlight that (i) using an objective function focused on similarity (WACH) and considering the nearest prototypical records (CEGP) helps in generating counterfactuals not too dissimilar from the instance under analysis; (ii) when the objective function is regularized (CEM) and the variations are controlled (BF) the counterfactuals generated differs from the instance under analysis by only a few attributes. On the other hand, concerning *diversity*, DICE and CBCE seems the best performer for both div_{dist} and div_{count}. In particular, DICE outperforms all the other methods for div_{count}, while CBCE is the best when $k \geq 10$.

Methods returning a single counterfactual have these measures equal to zero by default. If we consider simultaneously the plots on dissimilarity (dis_{dist} and dis_{count}) and diversity (div_{dist} and div_{count}), it turns out that BF is the method that has the best trade off and good values for both of them.

Regarding *implausibility* (*impl*), the best performers are the methods that return a single counterfactual, i.e., CEM, CEGP and WACH. This highlights the fact that when a single counterfactual is returned, it is more likely that it is not an outlier with respect to a reference population, and it resembles other existing instances. Thus, even though these approaches are exogenous, they are good in generating reliable counterfactuals. Among the methods returning more than a counterfactual CBCE is constantly smaller that DICE and BF with respect to implausibility. This is due to (i) the fact that CBCE is an endogenous explainer, (ii) the metric adopted to evaluate implausibility that is a "distance-based" one. Perhaps, the usage of metrics derived from outlier detection which are not dependent form the number of features, e.g., Isolation Forest (Liu et al. 2008), would have lowered the implausibility for CBCE. Also CEML, hidden behind CEM in the plot, has remarkable performance in terms of implausibility.

For the *discriminative power* (*dipo*), we observe opposite results with respect to implausibility. Indeed, the explainer generating counterfactual that helps in separating between classes are DICE, CBCE, BF and CEML. From this empirical comparison, we can notice that (i) explainers producing less plausible counterfactuals are those that can help more in differentiating between classes, (ii) CBCE achieves the best trade-off among these two aspects.

From the actionability (*act*) plot we observe that only BF and CBCE return a notable fraction of actionable counterfactuals. DICE and CEML return more than one counterfactual, and they allow specifying the actionable features, but they do not actually check their results for validity or actionability. Thus, the endogenous approach CBCE and the brute force BF testing alternatives of features are the best explainers accounting for actionability "by design".

Concerning instability $inst_{x,\bar{x}}$, CEML is the most stable, and CBCE the most unstable. For CEM, CEGP and WACH[15] we have to account that, in many cases, they do not return counterfactuals for two similar instances $(x, \bar{x})$, and therefore, the measure is biased. Concerning instability with respect to the same instance $inst_{x,x}$, DICE is the less stable method.

Finally, all the explainers, with the exception of BF, require on average a runtime of more than one minute. This result is in contrast to the expectations and to the discussion presented in Sect. 4.2. However, it is because the experimented explanation methods are tested with default parameter setting that allows BF to vary at most two features among the m available.

Figure 3 shows the performance of re-implemented counterfactual explainers not based on optimization strategies.[16] All these approaches return valid and actionable counterfactuals by design. GSG and NNCE almost always return all the counterfactuals required (*size*). On the other hand, TBCE- P and BF1 can also return less than half of the

[15] The high stability of WACH is due to the fact that the `alibi` library implementing it by default is initializing x' with x itself instead of with random samples.

[16] We highlight that our implementation of GSG is able to return more than a counterfactual by selecting the k nearest ones to x after that the algorithm stops.

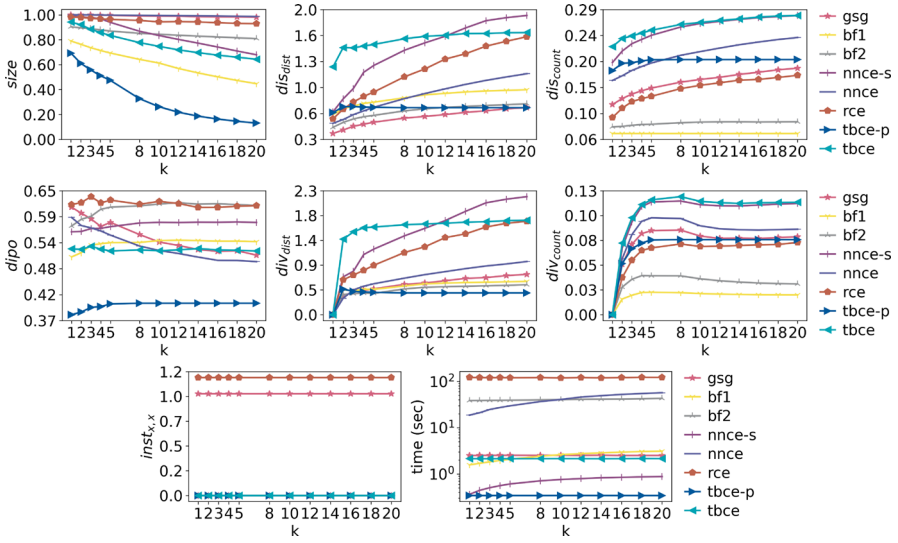

Fig. 3 Aggregated metrics for explainers not based on optimization strategies varying the number of counterfactuals k. Best view in color. (Color figure online)

counterfactuals required for values of $k > 15$. Concerning the dissimilarity we have the following results. Looking at dis_{dist}, GSG is the approach that returns counterfactuals more similar to x, yet changing more features than BF1 and BF2. Indeed, with respect to dis_{count} BF1 and BF2 have a lower number of changes with respect to x due to the fact that they have a strict upper bound for it. As one would expect, the metric slowly increases with k. On the other hand, concerning diversity, (div_{dist} and div_{count}), NNCE-S and TBCE are the best performers. This result is due to the fact that (i) the selection of feature values among the selected samples can induct a more variegate number of changes, and (ii) the usage of the tree forces the creation of different counterfactuals belonging to the leaves of the tree less populated. As for the results in Fig. 2, for *discriminative power* (*dipo*) shows opposite results with respect to implausibility (not reported here). All the explainers except TBCE- P have a *dipo* among 0.54 and 0.65, i.e., in line with those of the other explainers implemented in existing libraries. As a consequence, they all exhibit a comparable level of $inst_{x,\bar{x}}$ with TBCE- P being the most stable. With respect to instability for the same instance ($inst_{x,x}$) the less stable approaches are GSG and RCE, while all the others are consistently stable.[17] This result is due to the fact that among these approaches GSG and RCE are the only one making use of random procedures. Finally, the methods which are clearly less efficient, requiring on average more than twenty seconds, include RCE, BF2 and NNCE. In general, it is proved that counterfactual explainers not based on optimization strategies are faster than the others.

As the last experiment, in Fig. 4 we show how different distance functions can affect the performance of counterfactual explainers relying on a distance function. In particular, we focus on GSG and NNCE, and we experiment with the following mixed

[17] Same results are observed for $inst_{x,\bar{x}}$.

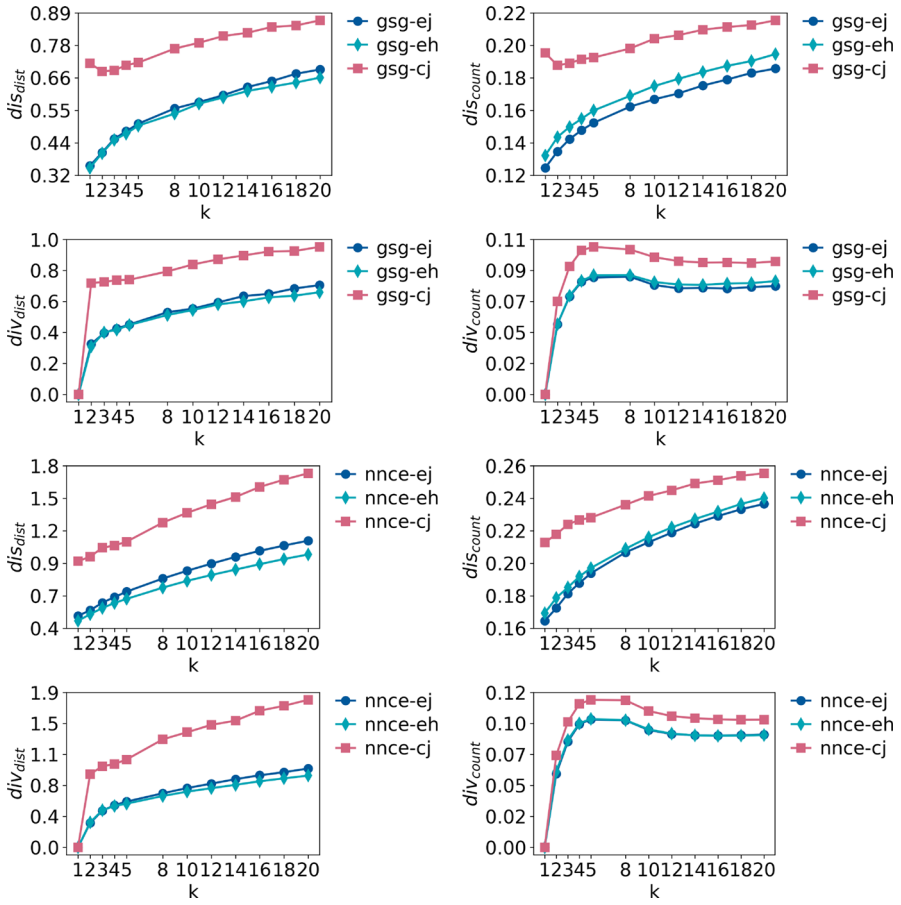

Fig. 4 Aggregated metrics for explainers not based on optimization strategies varying the number of required counterfactuals k and the distance function adopted. Best view in color. (Color figure online)

distance functions: Euclidean for continuous and Jaccard dissimilarity for categorical *(ej)*, used as default in the previous experiments, Euclidean for continuous and Hamming for categorical *(eh)*, and Cosine distance for continuous and Jaccard dissimilarity for categorical *(cj)*. We notice that *(ej)* mainly contributes to minimizing the number of features changed but, in general, has comparable results with *(eh)*. On the other hand, *(cj)* allows increasing the diversity at the cost of penalizing the dissimilarity. Therefore, the differences in the results for the different distance metrics seems to be mainly driven by the metric used for continuous features. As summary we can say that different distance functions can affect the result especially in terms of similarity or diversity. Aiming for a more "conservative" behavior that wants to find similar counterfactuals a user should probably select the Euclidean-Jaccard/Hamming distance.

The main findings of this benchmarking are the following. First, explainers searching for more than a counterfactual return at least a counterfactual more likely than

those returning a single instance. On the other hand, when a single counterfactual is returned, it is typically more plausible with respect to a reference population than many different instances. This result is also confirmed for exogenous approaches based on optimization strategies. Second, it seems that less plausible counterfactuals better helps in highlighting crucial aspects causing the change in the classification outcome. Third, methods using an objective function focused on similarity and considering prototypes produce counterfactuals more similar to the input instance, while methods using an objective function regularized can manage and prefer the generation of diverse counterfactuals. Fourth, endogenous approaches have the best trade-off between similarity, diversity, and plausibility. Fifth, explainers adopting random procedures and optimization strategies are more prone to instability. Sixth, methods not using optimization strategies are more efficient and better guarantee validity and actionability. Finally, explainers based on distance functions are subject to the choice of the distance function adopted that can push towards improving similarity rather than diversity or vice-versa.

7 Conclusion

We have presented a survey of the latest advances on counterfactual explanation methods by proposing a categorization based on the strategies adopted to retrieve the counterfactuals, on the properties guaranteed by the explanations, and on the characteristic of the method. Besides, we have measured and evaluated a set of counterfactual explainers with a quantitative comparison.

Our literature review reveals that there was an incredible increment in the development of counterfactual explanation methods in the last two years. Most of the counterfactual explainers adopt an optimization algorithm and try to insert into the loss function more and more penalization terms to control different desired properties for the counterfactuals returned. Moreover, most of these terms are designed to account for plausibility and often rely on pre-trained autoencoders to evaluate the generated examples. The main limitation of these methods is that they are typically model-specific and focus on returning a single counterfactual. On the other hand, counterfactual explainers using other techniques are generally model-agnostic and return a set of diverse counterfactuals. Nearly all the explainers guarantee validity and, when the one-hot encoding format is admitted and adequately managed, can handle categorical attributes. More or less half of the reviewed methods take into account actionability, while only six models also causality. Instance-based explainers guarantee plausibility returning endogenous counterfactuals, while all the others are exogenous and adopt different forms of penalization to account for it. From the experiments emerged that the counterfactual explainers that theoretically guarantee more properties are those who typically return a single counterfactual and require the highest computational time. Also, the explainers either provide very similar counterfactuals or a set of diverse counterfactuals which are necessarily less similar to the instance under analysis. Explainers based on optimization strategies are typically one or two orders of magnitude slower than those based on heuristics, instances, and decision trees. Moreover, the running time is not the only metric in favor of explainers not

based on optimization strategies. Finally, we observed that different distance functions can enhance a counterfactual search more focused on similarity or more focused on diversity. As a general recommendation, perhaps, we might suggest to the reader to experiment first with endogenous counterfactual explainers that seem to play at the intersection of the best performers with respect to all the proprieties benchmarked in our work.

In recent years, the contributions in XAI of counterfactual explainers have constantly grown. However, there are still a large number of research directions that need to be addressed. At the moment, there are no methods that account for all the properties and which are model and data agnostic. Causality is perhaps the property less managed and the one in which researchers in XAI should invest more time. Counterfactual explanations are local explanations. However, it would be interesting to understand if they can be merged to obtain a global explanation of the black-box model (Setzu et al. 2021). In addition, all the explainers analyzed are unable to handle missing attributes. Indeed, since they all rely on a notion of distance between instances, if the input instance has a missing value for a set of attributes, then the explainer cannot be applied without imputation even though the black-box classifier can deal with missing values. It is important to underline that a certain imputation rather than another one could affect the explanation process. With respect to evaluation, there is not yet a standard agreement on how to evaluate counterfactual explainers. Indeed, researchers currently adopt various measures which are sometimes not easy to compare among different papers. Furthermore, another trend of research could focus more on the human side, emphasizing the human-machine interactions in terms of counterfactuals. For instance, features changed by the counterfactuals could be distinguished between "foreground feature", i.e., those that the user is aware of, and "background features", i.e., those that the user is not aware of, or cannot consider due to its experience or existing limitations. In other words, the decision-making system could base its decisions on a set of hidden features from the user. Therefore, the related counterfactuals could either be incomplete because they do not consider the "background features", or not actionable because if the "background features" are not known, it would be difficult for the user to act to change them. For instance, in the loan request to a bank, an example of foreground features are the amount and duration requested, while we can consider other debts of relatives or friends of the applicant as background features. Another important factor that is not considered yet in the literature but highly involves humans is time. Indeed, a counterfactual could be valid at a certain time, but perhaps when the model is able to provide the recourse, then it is not valid anymore because the model was updated in the meantime. This can be due to some confounding factors that simultaneously changed as a consequence of existing causalities or to the passing of time. We believe that counterfactual explanation analysis must be addressed more in the development of AI applications in the future, and we hope that this survey could help in its development.

Acknowledgements This work has been partially supported by the European Community Horizon 2020 programme under the funding schemes: G.A. 871042 *SoBigData++* (sobigdata), G.A. 952026 *HumanE AI Net* (humane-ai), G.A. 834756 *XAI* (xai), and G.A. 952215 *TAILOR* (tailor).

Author Contributions RG: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data Curation, Writing—Original Draft, Writing—Review & Editing, Visualization.

Funding Open access funding provided by Universitá di Pisa within the CRUI-CARE Agreement.

Availability of data and materials The datasets adopted in this work are open source and available at https://archive.ics.uci.edu/ml/datasets.php, https://www.kaggle.com/datasets, https://community.fico.com/s/explainable-machine-learning-challenge.

Declarations

Conflict of interest The author declare that he has no conflict of interest.

Code availability The code is open source, and can be downloaded at https://github.com/riccotti/Scamander.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication The author declare that he provides consent for publication.

References

Aamodt A, Plaza E (1994) Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Commun 7(1):39–59

Adadi A et al (2018) Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access 6:52138–52160

Aggarwal CC, Chen C, Han J (2010) The inverse classification problem. J Comput Sci Technol 25(3):458–468

Anjomshoae S, Najjar A, Calvaresi D, Främling K (2019) Explainable agents and robots: results from a systematic literature review. In: Proceedings of the 18th international conference on autonomous agents and multiagent systems, AAMAS'19, Montreal, QC, Canada, May 13–17, 2019, International Foundation for Autonomous Agents and Multiagent Systems, pp 1078–1088

Arrieta AB, Ser JD (2020) Plausible counterfactuals: auditing deep learning classifiers with realistic adversarial examples. In: 2020 International joint conference on neural networks, IJCNN 2020, Glasgow, United Kingdom, July 19–24, 2020, IEEE, pp 1–7

Arrieta AB, Rodríguez ND, Ser JD, Bennetot A, Tabik S, Barbado A, García S, Gil-Lopez S, Molina D, Benjamins R, Chatila R, Herrera F (2020) Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf Fusion 58:82–115

Artelt A (2019) Ceml: counterfactuals for explaining machine learning models—a python toolbox. https://www.github.com/andreArtelt/ceml

Artelt A, Hammer B (2019) On the computation of counterfactual explanations—a survey. CoRR arXiv:1911.07749

Artelt A, Hammer B (2020a) Convex density constraints for computing plausible counterfactual explanations. In: Artificial neural networks and machine learning—ICANN 2020—29th international

conference on artificial neural networks, Bratislava, Slovakia, September 15–18, 2020, Proceedings, Part I, Springer, Lecture notes in computer science, vol 12396, pp 353–365
Artelt A, Hammer B (2020b) Efficient computation of counterfactual explanations of LVQ models. In: 28th European symposium on artificial neural networks, computational intelligence and machine learning, ESANN 2020, Bruges, Belgium, October 2–4, 2020, pp 19–24
Artelt A, Vaquet V, Velioglu R, Hinder F, Brinkrolf J, Schilling M, Hammer B (2021) Evaluating robustness of counterfactual explanations. CoRR arXiv:2103.02354
Ates E, Aksar B, Leung VJ, Coskun AK (2021) Counterfactual explanations for machine learning on multivariate time series data. In: 2021 international conference on applied artificial intelligence (ICAPAI), IEEE, pp 1–8
Balasubramanian R, Sharpe S, Barr B, Wittenbach JD, Bruss CB (2020) Latent-cf: a simple baseline for reverse counterfactual explanations. CoRR arXiv:2012.09301
Ballet V, Renard X, Aigrain J, Laugel T, Frossard P, Detyniecki M (2019) Imperceptible adversarial attacks on tabular data. CoRR arXiv:1911.03274
Barbaglia L, Manzan S, Tosetti E (2020) Forecasting loan default in Europe with machine learning. Available at SSRN 3605449
Beck A, Teboulle M (2009) A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J Imaging Sci 2(1):183–202
Beck SR, Riggs KJ, Gorniak SL (2009) Relating developments in children's counterfactual thinking and executive functions. Think Reason 15(4):337–354
Bhatt U, Xiang A, Sharma S, Weller A, Taly A, Jia Y, Ghosh J, Puri R, Moura JMF, Eckersley P (2020) Explainable machine learning in deployment. In: FAT*'20: conference on fairness, accountability, and transparency, Barcelona, Spain, January 27–30, 2020, ACM, pp 648–657
Bien J, Tibshirani R et al (2011) Prototype selection for interpretable classification. Ann Appl Stat 5(4):2403–2424
Bodria F, Giannotti F, Guidotti R, Naretto F, Pedreschi D, Rinzivillo S (2021) Benchmarking and survey of explanation methods for black box models. CoRR arXiv:2102.13076
Breunig MM, Kriegel H, Ng RT, Sander J (2000) LOF: identifying density-based local outliers. In: Proceedings of the 2000 ACM SIGMOD international conference on management of data, May 16–18, 2000, Dallas, Texas, USA, ACM, pp 93–104
Brughmans D, Martens D (2021) NICE: an algorithm for nearest instance counterfactual explanations. CoRR arXiv:2104.07411
Buchsbaum D, Bridgers S, Skolnick Weisberg D, Gopnik A (2012) The power of possibility: causal learning, counterfactual reasoning, and pretend play. Philos Trans R Soc B Biol Sci 367(1599):2202–2212
Byrne RMJ (2019) Counterfactuals in explainable artificial intelligence (XAI): evidence from human reasoning. In: Kraus S (ed) Proceedings of the twenty-eighth international joint conference on artificial intelligence, IJCAI 2019, Macao, China, August 10–16, 2019, ijcai.org, pp 6276–6282
Carlini N, Wagner DA (2017) Adversarial examples are not easily detected: bypassing ten detection methods. In: Proceedings of the 10th ACM workshop on artificial intelligence and security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017, ACM, pp 3–14
Carreira-Perpiñán MÁ, Hada SS (2021) Counterfactual explanations for oblique decision trees: exact, efficient algorithms. In: Thirty-Fifth AAAI conference on artificial intelligence, AAAI 2021, thirty-third conference on innovative applications of artificial intelligence, IAAI 2021, the eleventh symposium on educational advances in artificial intelligence, EAAI 2021, Virtual Event, February 2–9, 2021, AAAI Press, pp 6903–6911
Carvalho DV, Pereira EM, Cardoso JS (2019) Machine learning interpretability: a survey on methods and metrics. Electronics 8(8):832
Chapman-Rounds M, Schulz M, Pazos E, Georgatzis K (2019) EMAP: explanation by minimal adversarial perturbation. CoRR arXiv:1912.00872
Chapman-Rounds M, Bhatt U, Pazos E, Schulz M, Georgatzis K (2021) FIMAP: feature importance by minimal adversarial perturbation. In: Thirty-fifth AAAI conference on artificial intelligence, AAAI 2021, thirty-third conference on innovative applications of artificial intelligence, IAAI 2021, the eleventh symposium on educational advances in artificial intelligence, EAAI 2021, Virtual Event, February 2–9, 2021, AAAI Press, pp 11433–11441. https://ojs.aaai.org/index.php/AAAI/article/view/17362
Cheng F, Ming Y, Qu H (2021) DECE: decision explorer with counterfactual explanations for machine learning models. IEEE Trans Vis Comput Graph 27(2):1438–1447

Choi Y, Choi M, Kim M, Ha J, Kim S, Choo J (2018) Stargan: unified generative adversarial networks for multi-domain image-to-image translation. In: 2018 IEEE conference on computer vision and pattern recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–22, 2018, Computer Vision Foundation/IEEE Computer Society, pp 8789–8797

Craven MW et al (1995) Extracting tree-structured representations of trained networks. In: Touretzky DS, Mozer M, Hasselmo ME (eds) Advances in neural information processing systems 8, NIPS, Denver, CO, USA, November 27–30, 1995. MIT Press, pp 24–30

Cui Z, Chen W, He Y, Chen Y (2015) Optimal action extraction for random forests and boosted trees. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, Sydney, NSW, Australia, August 10–13, 2015, ACM, pp 179–188

Dandl S, Molnar C, Binder M, Bischl B (2020) Multi-objective counterfactual explanations. In: Parallel problem solving from nature - PPSN XVI - 16th international conference, PPSN 2020, Leiden, The Netherlands, September 5–9, 2020, Proceedings, Part I, Springer, Lecture notes in computer science, vol 12269, pp 448–469

Dhurandhar A, Chen P, Luss R, Tu C, Ting P, Shanmugam K, Das P (2018) Explanations based on the missing: towards contrastive explanations with pertinent negatives. In: Advances in neural information processing systems 31: annual conference on neural information processing systems 2018, NeurIPS 2018, December 3–8, 2018, Montréal, Canada, pp 590–601

Dhurandhar A, Pedapati T, Balakrishnan A, Chen P, Shanmugam K, Puri R (2019) Model agnostic contrastive explanations for structured data. CoRR arXiv:1906.00117

Dosilovic FK, Brcic M, Hlupic N (2018) Explainable artificial intelligence: a survey. In: 41st international convention on information and communication technology, electronics and microelectronics, MIPRO 2018, Opatija, Croatia, May 21–25, 2018, IEEE, pp 210–215

Downs M, Chu JL, Yacoby Y, Doshi-Velez F, Pan W (2020) CRUDS: counterfactual recourse using disentangled subspaces. In: ICML workshop on human interpretability in machine learning

Fan C, Li P (2020) Classification acceleration via merging decision trees. In: FODS'20: ACM-IMS foundations of data science conference, virtual event, USA, October 19–20, 2020, ACM, pp 13–22

Fernandez C, Provost FJ, Han X (2020) Explaining data-driven decisions made by AI systems: the counterfactual approach. CoRR arXiv:2001.07417

Fernández RR, de Diego IM, Aceña V, Fernández-Isabel A, Moguerza JM (2020) Random forest explainability using counterfactual sets. Inf Fusion 63:196–207

Freitas AA (2013) Comprehensible classification models: a position paper. SIGKDD Explor 15(1):1–10

Ghazimatin A, Balalau O, Roy RS, Weikum G (2020) PRINCE: provider-side interpretability with counterfactual explanations in recommender systems. In: WSDM'20: the thirteenth ACM international conference on web search and data mining, Houston, TX, USA, February 3–7, 2020, ACM, pp 196–204

Gilpin LH, Bau D, Yuan BZ, Bajwa A, Specter M, Kagal L (2018) Explaining explanations: an overview of interpretability of machine learning. In: 5th IEEE international conference on data science and advanced analytics, DSAA 2018, Turin, Italy, October 1–3, 2018, IEEE, pp 80–89

Goebel R, Chander A, Holzinger K, Lécué F, Akata Z, Stumpf S, Kieseberg P, Holzinger A (2018) Explainable AI: the new 42? In: Machine learning and knowledge extraction - second IFIP TC 5, TC 8/WG 8.4, 8.9, TC 12/WG 12.9 International cross-domain conference, CD-MAKE 2018, Hamburg, Germany, August 27–30, 2018, Proceedings, Springer, Lecture notes in computer science, vol 11015, pp 295–303

Gomez O, Holter S, Yuan J, Bertini E (2020) Vice: visual counterfactual explanations for machine learning models. In: IUI'20: 25th international conference on intelligent user interfaces, Cagliari, Italy, March 17–20, 2020, ACM, pp 531–535

Goyal Y, Wu Z, Ernst J, Batra D, Parikh D, Lee S (2019) Counterfactual visual explanations. In: Proceedings of the 36th international conference on machine learning, ICML 2019, 9–15 June 2019, Long Beach, California, USA, PMLR, Proceedings of machine learning research, vol 97, pp 2376–2384

Guidotti R (2021) Evaluating local explanation methods on ground truth. Artif Intell 291:103428

Guidotti R, Monreale A (2020) Data-agnostic local neighborhood generation. In: 20th IEEE international conference on data mining, ICDM 2020, Sorrento, Italy, November 17–20, 2020, IEEE, pp 1040–1045

Guidotti R, Ruggieri S (2019) On the stability of interpretable models. In: International joint conference on neural networks, IJCNN 2019 Budapest, Hungary, July 14–19, 2019, IEEE, pp 1–8

Guidotti R, Monreale A, Giannotti F, Pedreschi D, Ruggieri S, Turini F (2019) Factual and counterfactual explanations for black box decision making. IEEE Intell Syst 34(6):14–23

Guidotti R, Monreale A, Matwin S, Pedreschi D (2019b) Black box explanation by learning image exemplars in the latent feature space. In: Machine learning and knowledge discovery in databases—European conference, ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part I, Springer, Lecture notes in computer science, vol 11906, pp 189–205

Guidotti R, Monreale A, Ruggieri S, Turini F, Giannotti F, Pedreschi D (2019) A survey of methods for explaining black box models. ACM Comput Surv 51(5):93:1-93:42

Guidotti R, Monreale A, Spinnato F, Pedreschi D, Giannotti F (2020) Explaining any time series classifier. In: 2nd IEEE international conference on cognitive machine intelligence, CogMI 2020, Atlanta, GA, USA, October 28–31, 2020, IEEE, pp 167–176

Hashemi M, Fathi A (2020) Permuteattack: counterfactual explanation of machine learning credit scorecards. CoRR arXiv:2008.10138

He Z, Zuo W, Kan M, Shan S, Chen X (2019) Attgan: facial attribute editing by only changing what you want. IEEE Trans Image Process 28(11):5464–5478. https://doi.org/10.1109/TIP.2019.2916751

Joshi S, Koyejo O, Vijitbenjaronk W, Kim B, Ghosh J (2019) Towards realistic individual recourse and actionable explanations in black-box decision making systems. CoRR arXiv:1907.09615

Kanamori K, Takagi T, Kobayashi K, Arimura H (2020) DACE: distribution-aware counterfactual explanation by mixed-integer linear optimization. In: Proceedings of the twenty-ninth international joint conference on artificial intelligence, IJCAI 2020, ijcai.org, pp 2855–2862

Kanamori K, Takagi T, Kobayashi K, Ike Y, Uemura K, Arimura H (2021) Ordered counterfactual explanation by mixed-integer linear optimization. In: Thirty-Fifth AAAI conference on artificial intelligence, AAAI 2021, thirty-third conference on innovative applications of artificial intelligence, IAAI 2021, the eleventh symposium on educational advances in artificial intelligence, EAAI 2021, Virtual Event, February 2–9, 2021, AAAI Press, pp 11564–11574

Kang S, Jung H, Won D, Lee S (2020) Counterfactual explanation based on gradual construction for deep networks. CoRR arXiv:2008.01897

Karimi A, Barthe G, Balle B, Valera I (2020) Model-agnostic counterfactual explanations for consequential decisions. In: The 23rd international conference on artificial intelligence and statistics, AISTATS 2020, 26–28 August 2020, Online [Palermo, Sicily, Italy], PMLR, Proceedings of machine learning research, vol 108, pp 895–905

Karimi A, Barthe G, Schölkopf B, Valera I (2021a) A survey of algorithmic recourse: definitions, formulations, solutions, and prospects. CoRR arXiv:2010.04050

Karimi A, Schölkopf B, Valera I (2021b) Algorithmic recourse: from counterfactual explanations to interventions. In: FAccT'21: 2021 ACM conference on fairness, accountability, and transparency, virtual event/Toronto, Canada, March 3–10, 2021, ACM, pp 353–362

Keane MT, Smyth B (2020) Good counterfactuals and where to find them: a case-based technique for generating counterfactuals for explainable AI (XAI). In: Case-based reasoning research and development—28th international conference, ICCBR 2020, Salamanca, Spain, June 8–12, 2020, Proceedings, Springer, Lecture notes in computer science, vol 12311, pp 163–178

Keane MT, Kenny EM, Delaney E, Smyth B (2021) If only we had better counterfactual explanations: five key deficits to rectify in the evaluation of counterfactual XAI techniques. In: Proceedings of the thirtieth international joint conference on artificial intelligence, IJCAI 2021, Virtual Event/Montreal, Canada, 19–27 August 2021, ijcai.org, pp 4466–4474

Kenny EM, Keane MT (2021) On generating plausible counterfactual and semi-factual explanations for deep learning. In: Thirty-fifth AAAI conference on artificial intelligence, AAAI 2021, thirty-third conference on innovative applications of artificial intelligence, IAAI 2021, the eleventh symposium on educational advances in artificial intelligence, EAAI 2021, Virtual Event, February 2–9, 2021, AAAI Press, pp 11575–11585

Kianpour M, Wen S (2019) Timing attacks on machine learning: state of the art. In: Intelligent systems and applications - proceedings of the 2019 intelligent systems conference, IntelliSys 2019, London, UK, September 5–6, 2019, Volume 1, Springer, Advances in intelligent systems and computing, vol 1037, pp 111–125

Kim B, Koyejo O, Khanna R (2016) Examples are not enough, learn to criticize! criticism for interpretability. In: Advances in neural information processing systems 29: annual conference on neural information processing systems 2016, December 5–10, 2016, Barcelona, Spain, pp 2280–2288

Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: 3rd international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference track proceedings

Klys J, Snell J, Zemel RS (2018) Learning latent subspaces in variational autoencoders. In: Advances in neural information processing systems 31: annual conference on neural information processing systems 2018, NeurIPS 2018, December 3–8, 2018, Montréal, Canada, pp 6445–6455

Koh PW, et al. (2017) Understanding black-box predictions via influence functions. In: Proceedings of the 34th international conference on machine learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017, PMLR, Proceedings of machine learning research, vol 70, pp 1885–1894

Kovalev M, Utkin LV, Coolen FPA, Konstantinov AV (2021) Counterfactual explanation of machine learning survival models. Informatica 32(4):817–847

Kusner MJ, Loftus JR, Russell C, Silva R (2017) Advances in neural information processing systems 30: annual conference on neural information processing systems 2017, December 4–9, 2017, long beach, ca, USA. In: NIPS, pp 4066–4076

Lampridis O, Guidotti R, Ruggieri S (2020) Explaining sentiment classification with synthetic exemplars and counter-exemplars. In: Discovery science—23rd international conference, DS 2020, Thessaloniki, Greece, October 19–21, 2020, Proceedings, Springer, Lecture notes in computer science, vol 12323, pp 357–373

Lash MT, Lin Q, Street WN, Robinson JG (2017a) A budget-constrained inverse classification framework for smooth classifiers. In: 2017 IEEE international conference on data mining workshops, ICDM workshops 2017, New Orleans, LA, USA, November 18–21, 2017, IEEE Computer Society, pp 1184–1193

Lash MT, Lin Q, Street WN, Robinson JG, Ohlmann JW (2017b) Generalized inverse classification. In: Proceedings of the 2017 SIAM international conference on data mining, Houston, Texas, USA, April 27–29, 2017, SIAM, pp 162–170

Laugel T, Lesot M, Marsala C, Renard X, Detyniecki M (2018) Comparison-based inverse classification for interpretability in machine learning. In: Information processing and management of uncertainty in knowledge-based systems. Theory and foundations—17th international conference, IPMU 2018, Cádiz, Spain, June 11–15, 2018, Proceedings, Part I, Springer, Communications in computer and information science, vol 853, pp 100–111

Laugel T, Lesot M, Marsala C, Renard X, Detyniecki M (2019) The dangers of post-hoc interpretability: unjustified counterfactual explanations. In: Proceedings of the twenty-eighth international joint conference on artificial intelligence, IJCAI 2019, Macao, China, August 10–16, 2019, ijcai.org, pp 2801–2807

Le T, Wang S, Lee D (2020) GRACE: generating concise and informative contrastive sample to explain neural network model's prediction. In: KDD'20: the 26th ACM SIGKDD conference on knowledge discovery and data mining, Virtual Event, CA, USA, August 23–27, 2020, ACM, pp 238–248

Lee J, Mirrokni VS, Nagarajan V, Sviridenko M (2009) Non-monotone submodular maximization under matroid and knapsack constraints. In: Proceedings of the 41st annual ACM symposium on theory of computing, STOC 2009, Bethesda, MD, USA, May 31–June 2, 2009, ACM, pp 323–332

Li XH, Cao CC, Shi Y, Bai W, Gao H, Qiu L, Wang C, Gao Y, Zhang S, Xue X, et al (2020) A survey of data-driven and knowledge-aware explainable AI. IEEE Trans Knowl Data Eng

Lipton P (1990) Contrastive explanation. R Inst Philos Suppl 27:247–266

Liu FT, Ting KM, Zhou Z (2008) Isolation forest. In: Proceedings of the 8th IEEE international conference on data mining (ICDM 2008), December 15–19, 2008, Pisa, Italy, IEEE Computer Society, pp 413–422

Lucic A, Oosterhuis H, Haned H, de Rijke M (2019) Focus: flexible optimizable counterfactual explanations for tree ensembles. CoRR arXiv:1911.12199

Lucic A, Haned H, de Rijke M (2020) Why does my model fail? Contrastive local explanations for retail forecasting. In: FAT*'20: conference on fairness, accountability, and transparency, Barcelona, Spain, January 27–30, 2020, ACM, pp 90–98

Lucic A, Ter Hoeve M, Tolomei G, de Rijke M, Silvestri F (2021) Cf-gnnexplainer: counterfactual explanations for graph neural networks. CoRR arXiv:2102.03322

Lundberg SM, Lee S (2017) A unified approach to interpreting model predictions. In: Advances in neural information processing systems 30: annual conference on neural information processing systems 2017, December 4–9, 2017, Long Beach, CA, USA, pp 4765–4774

Mahajan D, Tan C, Sharma A (2019) Preserving causal constraints in counterfactual explanations for machine learning classifiers. CoRR arXiv:1912.03277

Martens D, Provost FJ (2014) Explaining data-driven document classifications. MIS Q 38(1):73–99

Martens D, Baesens B, Van Gestel T, Vanthienen J (2007) Comprehensible credit scoring models using rule extraction from support vector machines. Eur J Oper Res 183(3):1466–1476

Mazzine R, Martens D (2021) A framework and benchmarking study for counterfactual generating methods on tabular data. Appl Sci 11(16):7274
Mc Grath R, Costabello L, Le Van C, Sweeney P, Kamiab F, Shen Z, Lécué F (2018) Interpretable credit application predictions with counterfactual explanations. CoRR arXiv:1811.05245
McGill AL et al (1993) Contrastive and counterfactual reasoning in causal judgment. J Person Soc Psychol 64(6):897
Miller T (2018) Contrastive explanation: a structural-model approach. CoRR arXiv:1811.03163
Miller T (2019) Explanation in artificial intelligence: insights from the social sciences. Artif Intell 267:1–38
Mohammadi K, Karimi A, Barthe G, Valera I (2021) Scaling guarantees for nearest counterfactual explanations. In: AIES'21: AAAI/ACM conference on AI, ethics, and society, Virtual Event, USA, May 19–21, 2021, ACM, pp 177–187
Molnar C (2020) Interpretable machine learning. Lulu. com
Moore J, Hammerla N, Watkins C (2019) Explaining deep learning models with constrained adversarial examples. In: PRICAI 2019: trends in artificial intelligence—16th Pacific rim international conference on artificial intelligence, Cuvu, Yanuca Island, Fiji, August 26–30, 2019, Proceedings, Part I, Springer, Lecture notes in computer science, vol 11670, pp 43–56
Mothilal RK, Sharma A, Tan C (2020) Explaining machine learning classifiers through diverse counterfactual explanations. In: FAT*'20: conference on fairness, accountability, and transparency, Barcelona, Spain, January 27–30, 2020, ACM, pp 607–617
Mothilal RK, Mahajan D, Tan C, Sharma A (2021) Towards unifying feature attribution and counterfactual explanations: different means to the same end. In: AIES'21: AAAI/ACM conference on AI, ethics, and society, virtual event, USA, May 19–21, 2021, ACM, pp 652–663
Murdoch WJ, Singh C, Kumbier K, Abbasi-Asl R, Yu B (2019) Definitions, methods, and applications in interpretable machine learning. Proc Natl Acad Sci 116(44):22071–22080
Nebro AJ, Durillo JJ, García-Nieto J, Coello CAC, Luna F, Alba E (2009) SMPSO: a new pso-based metaheuristic for multi-objective optimization. In: 2009 IEEE symposium on computational intelligence in multi-criteria decision-making, MCDM 2009, Nashville, TN, USA, March 30–April 2, 2009, IEEE, pp 66–73
Numeroso D, Bacciu D (2021) MEG: generating molecular counterfactual explanations for deep graph networks. In: 2021 international joint conference on neural networks (IJCNN), IEEE, pp 1–8
Panigutti C, Perotti A, Pedreschi D (2020) Doctor XAI: an ontology-based approach to black-box sequential data classification explanations. In: FAT*'20: conference on fairness, accountability, and transparency, Barcelona, Spain, January 27–30, 2020, ACM, pp 629–639
Parmentier A, Vidal T (2021) Optimal counterfactual explanations in tree ensembles. In: Proceedings of the 38th international conference on machine learning, ICML 2021, 18–24 July 2021, Virtual Event, PMLR, Proceedings of machine learning research, vol 139, pp 8422–8431
Pawelczyk M, Broelemann K, Kasneci G (2020) Learning model-agnostic counterfactual explanations for tabular data. In: WWW'20: the web conference 2020, Taipei, Taiwan, April 20–24, 2020, ACM / IW3C2, pp 3126–3132
Pawelczyk M, Bielawski S, van den Heuvel J, Richter T, Kasneci G (2021) CARLA: a python library to benchmark algorithmic recourse and counterfactual explanation algorithms. CoRR arXiv:2108.00783
Pearl J et al (2009) Causal inference in statistics: an overview. Stat Surv 3:96–146
Powell MJD (1973) On search directions for minimization algorithms. Math Program 4(1):193–201
Poyiadzi R, Sokol K, Santos-Rodríguez R, De Bie T, Flach PA (2020) FACE: feasible and actionable counterfactual explanations. In: AIES'20: AAAI/ACM conference on AI, ethics, and society, New York, NY, USA, February 7–8, 2020, ACM, pp 344–350
Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I et al (2019) Language models are unsupervised multitask learners. OpenAI Blog 1(8):9
Ramakrishnan G, Lee YC, Albarghouthi A (2020) Synthesizing action sequences for modifying model decisions. In: The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, the thirty-second innovative applications of artificial intelligence conference, IAAI 2020, The tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020, New York, NY, USA, February 7–12, 2020, AAAI Press, pp 5462–5469
Ramon Y, Martens D, Provost FJ, Evgeniou T (2020) A comparison of instance-level counterfactual explanation algorithms for behavioral and textual data: Sedc, LIME-C and SHAP-C. Adv Data Anal Classif 14(4):801–819

Rathi S (2019) Generating counterfactual and contrastive explanations using SHAP. CoRR arXiv:1906.09293
Rawal K, Lakkaraju H (2020) Beyond individualized recourse: interpretable and interactive summaries of actionable recourses. In: Beyond individualized recourse: interpretable and interactive summaries of actionable recourses
Ribeiro MT, Singh S, Guestrin C (2016) "why should I trust you?": Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco, CA, USA, August 13–17, 2016, ACM, pp 1135–1144
Ribeiro MT, Singh S, Guestrin C (2018) Anchors: high-precision model-agnostic explanations. In: Proceedings of the thirty-second AAAI conference on artificial intelligence, (AAAI-18), the 30th innovative applications of artificial intelligence (IAAI-18), and the 8th AAAI symposium on educational advances in artificial intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2–7, 2018, AAAI Press, pp 1527–1535
Rockoff JE, Jacob BA, Kane TJ, Staiger DO (2011) Can you recognize an effective teacher when you recruit one? Educ Finance Policy 6(1):43–74
Russell C (2019) Efficient search for diverse coherent explanations. In: Proceedings of the conference on fairness, accountability, and transparency, FAT* 2019, Atlanta, GA, USA, January 29–31, 2019, ACM, pp 20–28
Samek W, Montavon G, Vedaldi A, Hansen LK, Müller K (eds) (2019) Explainable AI: interpreting, explaining and visualizing deep learning, Lecture notes in computer science, vol 11700. Springer
Schleich M, Geng Z, Zhang Y, Suciu D (2021) Geco: quality counterfactual explanations in real time. Proc VLDB Endow 14(9):1681–1693
Setzu M, Guidotti R, Monreale A, Turini F, Pedreschi D, Giannotti F (2021) Glocalx—from local to global explanations of black box AI models. Artif Intell 294:103457
Shakhnarovich G, Darrell T, Indyk P (2008) Nearest-neighbor methods in learning and vision. IEEE Trans Neural Netw 19(2):377
Sharma S, Henderson J, Ghosh J (2019) CERTIFAI: counterfactual explanations for robustness, transparency, interpretability, and fairness of artificial intelligence models. CoRR arXiv:1905.07857
Slack D, Hilgard S, Lakkaraju H, Singh S (2021) Counterfactual explanations can be manipulated. Advances in Neural Information Processing Systems 34
Sokol K, Santos-Rodríguez R, Flach PA (2019) FAT forensics: a python toolbox for algorithmic fairness, accountability and transparency. CoRR arXiv:1909.05167
Stepin I, Alonso JM, Catalá A, Pereira-Fariña M (2021) A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence. IEEE Access 9:11974–12001
Strecht P (2015) A survey of merging decision trees data mining approaches. In: Proceedings of 10th doctoral symposium in informatics engineering, pp 36–47
Tjoa E, Guan C (2019) A survey on explainable artificial intelligence (XAI): towards medical XAI. CoRR arXiv:1907.07374
Tolomei G, Silvestri F, Haines A, Lalmas M (2017) Interpretable predictions of tree-based ensembles via actionable feature tweaking. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, Halifax, NS, Canada, August 13–17, 2017, ACM, pp 465–474
Tomsett R, Braines D, Harborne D, Preece AD, Chakraborty S (2018) Interpretable to whom? A role-based model for analyzing interpretable machine learning systems. CoRR arXiv:1806.07552
Tsirtsis S, Rodriguez MG (2020) Decisions, counterfactual explanations and strategic behavior. In: NeurIPS
Ustun B, Spangher A, Liu Y (2019) Actionable recourse in linear classification. In: Proceedings of the conference on fairness, accountability, and transparency, FAT* 2019, Atlanta, GA, USA, January 29–31, 2019, ACM, pp 10–19
Van Der Waa J, Robeer M, Van Diggelen J, Brinkhuis M, Neerincx MA (2019) Contrastive explanations with local foil trees. CoRR arXiv:1806.07470
Van Looveren A, Klaise J (2021) Interpretable counterfactual explanations guided by prototypes. In: Machine learning and knowledge discovery in databases. Research Track - European conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part II, Springer, Lecture notes in computer science, vol 12976, pp 650–665
Verma S, Dickerson JP, Hines K (2020) Counterfactual explanations for machine learning: a review. CoRR arXiv:2010.10596
Vermeire T, Martens D (2022) Explainable image classification with evidence counterfactual. Pattern Analysis and Applications, pp 1–21

Visani G, Bagli E, Chesani F, Poluzzi A, Capuzzo D (2020) Statistical stability indices for LIME: obtaining reliable explanations for machine learning models. CoRR arXiv:2001.11757

Von Kügelgen J, Bhatt U, Karimi A, Valera I, Weller A, Schölkopf B (2020) On the fairness of causal algorithmic recourse. CoRR arXiv:2010.06529

Wachter S, Mittelstadt BD, Russell C (2017) Counterfactual explanations without opening the black box: automated decisions and the GDPR. Harv JL Tech 31:841

Wang P, Vasconcelos N (2020) SCOUT: self-aware discriminant counterfactual explanations. In: 2020 IEEE/CVF conference on computer vision and pattern recognition, CVPR 2020, Seattle, WA, USA, June 13–19, 2020, Computer Vision Foundation/IEEE, pp 8978–8987

Waters A, Miikkulainen R (2014) GRADE: machine learning support for graduate admissions. AI Mag 35(1):64–75

Wexler J, Pushkarna M, Bolukbasi T, Wattenberg M, Viégas FB, Wilson J (2020) The what-if tool: interactive probing of machine learning models. IEEE Trans Vis Comput Graph 26(1):56–65

White A, d'Avila Garcez AS (2020) Measurable counterfactual local explanations for any classifier. In: ECAI 2020—24th European conference on artificial intelligence, 29 August–8 September 2020, Santiago de Compostela, Spain, August 29–September 8, 2020 - Including 10th conference on prestigious applications of artificial intelligence (PAIS 2020), IOS Press, Frontiers in Artificial Intelligence and Applications, vol 325, pp 2529–2535

Wilson DR, Martinez TR (1997) Improved heterogeneous distance functions. J Artif Intell Res 6:1–34

Wu T, Ribeiro MT, Heer J, Weld DS (2021) Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In: Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing, ACL/IJCNLP 2021, Volume 1: Long Papers, virtual event, August 1–6, 2021, Association for Computational Linguistics, pp 6707–6723

Yang L, Kenny EM, Ng TLJ, Yang Y, Smyth B, Dong R (2020) Generating plausible counterfactual explanations for deep transformers in financial text classification. In: Proceedings of the 28th international conference on computational linguistics, COLING 2020, Barcelona, Spain (Online), December 8–13, 2020, International Committee on Computational Linguistics, pp 6150–6160

Zhang Y, Chen X (2020) Explainable recommendation: a survey and new perspectives. Found Trends Inf Retr 14(1):1–101

Zhang X, Solar-Lezama A, Singh R (2018) Interpreting neural network judgments via minimal, stable, and symbolic corrections. In: Advances in neural information processing systems 31: annual conference on neural information processing systems 2018, NeurIPS 2018, December 3–8, 2018, Montréal, Canada, pp 4879–4890

Zhao Y (2020) Fast real-time counterfactual explanations. CoRR arXiv:2007.05684

Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. J R Stat Soc Ser B Stat Methodol 67(2):301–320