Natural Example-Based Explainability: A Survey

Antonin Poché[1,2(✉)], Lucas Hervier[1,2], and Mohamed-Chafik Bakkay[1,2]

[1] IRT Saint Exupéry, Toulouse, France
{antonin.poche,lucas.hervier,mohamed-chafik.bakkay}@irt-saintexupery.com
[2] IRT SystemX, 2 boulevard Thomas Gobert, 91120 Palaiseau, France
{antonin.poche,lucas.hervier,mohamed-chafik.bakkay}@irt-systemx.fr

Abstract. Explainable Artificial Intelligence (XAI) has become increasingly significant for improving the interpretability and trustworthiness of machine learning models. While saliency maps have stolen the show for the last few years in the XAI field, their ability to reflect models' internal processes has been questioned. Although less in the spotlight, example-based XAI methods have continued to improve. It encompasses methods that use examples as explanations for a machine learning model's predictions. This aligns with the psychological mechanisms of human reasoning and makes example-based explanations natural and intuitive for users to understand. Indeed, humans learn and reason by forming mental representations of concepts based on examples.

This paper provides an overview of the state-of-the-art in natural example-based XAI, describing the pros and cons of each approach. A "natural" example simply means that it is directly drawn from the training data without involving any generative process. The exclusion of methods that require generating examples is justified by the need for plausibility which is in some regards required to gain a user's trust. Consequently, this paper will explore the following family of methods: similar examples, counterfactual and semi-factual, influential instances, prototypes, and concepts. In particular, it will compare their semantic definition, their cognitive impact, and added values. We hope it will encourage and facilitate future work on natural example-based XAI.

Keywords: Explainability · XAI · Survey · Example-based · Case-based · Counterfactuals · Semi-factuals · Influence Functions · Prototypes · Concepts

1 Introduction

With the ever-growing complexity of machine learning models and their large diffusion, understanding models' decisions and behavior became a necessity. Therefore, explainable artificial intelligence (XAI), the field that aims to understand

A. Poché and L. Hervier—These authors contributed equally to this work.

L. Longo (Ed.): xAI 2023, CCIS 1902, pp. 24–47, 2023.
https://doi.org/10.1007/978-3-031-44067-0_2

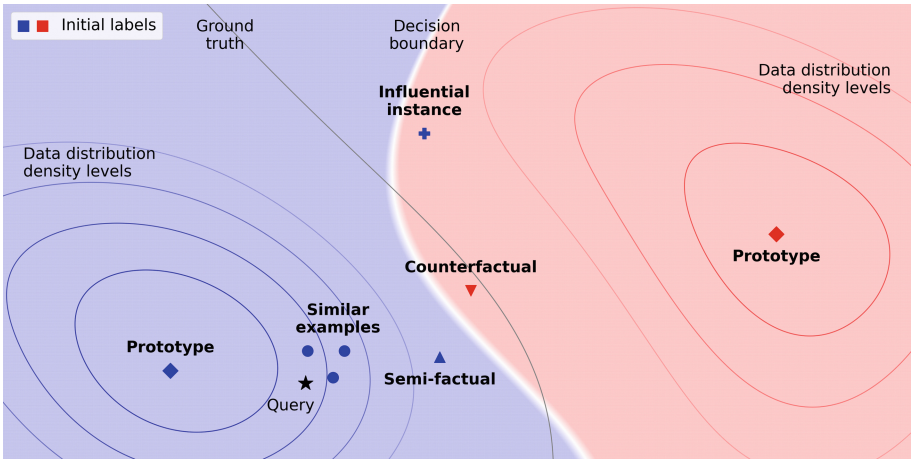

Fig. 1. Natural example-based explanation formats with respect to the query and the decision boundary. We can see similar examples are the closest elements to the query, while counterfactuals and semi-factuals are on either side of the point of the decision boundary the closest to the query. Prototypes are representative of each class in a dense zone of the dataset and the influential instance bends the decision boundary.

and clarify models, flourished with a huge diversity of methods. Several taxonomies have been proposed to differentiate between methods, with common components identified [2,4,50]: i) Local vs global: Local methods explain specific model decisions (in this case, the model's input is called the studied sample or query), while global methods provide insight into overall model behavior. ii) Post-hoc vs intrinsic vs explainable by-design: Post-hoc methods are applied to trained models, while by-design methods produce inherently explainable models. Intrinsic methods take into account model training without affecting the final state. iii) Black-box vs white-box: White-box methods require access to model weights/gradients. iv) Explanation formats which include: attribution methods [33,104], concepts [35,66], surrogate models [67,96], rule-based explanations [114], natural language explanations [19], dependencies [40,49], and example-based explanations [57,113].

Nonetheless, no matter the taxonomy of a method, its explanations are aimed at humans, hence, they should exploit the vast literature in philosophy, psychology, and cognitive science on how humans generate, understand, and react to explanations [79]. The psychology literature argued that, in everyday life, humans use examples as references to understand, explain something, or demonstrate their arguments [17,32,38,79,100]. Afterward, through user studies in the XAI field [35,51,61], researchers validated that example-based explainability provides better explanations over several other formats where example-based XAI corresponds to a family of methods where explanations are represented by or communicated through samples, or part of samples like crops.

However, previous surveying works on example-based XAI are either cursory as they survey XAI in general [2,4] or focus on a specific subset such as factual

methods [26,28,102] or contrastive explanations [59,84,113]. In fact, example-based explainability can be divided into several sub-formats with many similarities. As such, covering them together allows conclusions from sub-fields of the literature to serve one another. Thus, we believe a single work thoroughly mapping, describing, and analyzing each example-based XAI sub-format will benefit the field. Besides, this survey will only cover natural example-based explainability methods – *i.e* methods where examples are training samples and are not generated. Indeed, to generate high-dimensional data points, methods essentially rely on deep neural networks [6,62]. Nevertheless, for most high dimensional data, such approaches fail to ensure that generated examples are plausible and belong to the manifold (subspace of the input space where samples follow the data distribution), and examples need to be realistic for humans to interpret them [18]. Therefore, natural examples have two advantages, they do not use a model to explain another model which eases their acceptance, and natural examples are plausible by definition. In addition, apart from formats with only generative methods (such as feature visualizations [91]), we do not set aside any formats of example-based XAI as they may all bring new perspectives to others. Lastly, to navigate through the different formats we use the semantic definition of each format as it highlights the differences between formats. In some cases, examples from different formats may be the same sample, hence, clear semantic definitions are necessary to interpret examples.

Explanations in example-based explainability are all data points but there exist different semantic meanings to a given example. Depending on the relation between the example, the query, and the model, the information provided by the example will differ. The semantic definition of an example and the kind of insight it provides divide the example-based format into sub-groups, which are presented in Fig. 1. This overview is organized around those sub-groups (also called formats), this work will unfold as follows:

The first format is **similar examples** (or factuals) (Sect. 2), for the model, they are the closest elements to the query. Factuals give confidence in the prediction or explain misclassification, but they are limited to the close range of the considered sample. To provide insight into the model behavior on a larger zone around the query, **counterfactuals** and **semi-factuals** (Sects. 3.1 and 3.2) are more adapted. They are respectively the closest and the farthest samples on which the model makes a different and similar prediction. They are mainly used in classification, give insight into the decision boundary, and are complementary if paired. While they give an idea of the limit, they do not provide insights on how one could bend the decision boundaries of the model by altering the training data. This is addressed through **influential instances** (Sect. 4), the training samples with the highest impact on the model's state. In addition, contrary to previously listed example-based formats, influential instances are not limited to local explanations. Indeed, one can extract the most influential instances for the model in general. Another global explanation format is **Prototypes** (Sect. 5), which are a set of samples representative of either the dataset or a class. Most of the time they are selected without relying on the model and give an overview of the dataset, but some models are designed through prototypes, thus explainable

by design. **Concepts** (Sect. 6), a closely-related format, is also investigated. A concept is the abstraction of the common elements between samples – e.g. for trees, the concepts could be trunk, branch, and leaf. To communicate such concepts, if they are not labeled, the easiest way is through examples of such concepts (often part of samples such as patches).

Thus we could summarize the contributions of this paper as follows: i) To the best of our knowledge, we are the first to compile natural example-based explainability literature in a survey. Previous works either covered the whole XAI literature with a superficial analysis of example-based XAI or focused on a given sub-format of example-based XAI. ii) For each format we provide simple definitions, semantic meaning, key methods, their comparison, their pros and cons, and examples, and pros and cons. We additionally ground formats into social sciences and depict their cognitive added values when possible. iii) We explore, classify, and describe available methods in each natural example-based XAI format. We highlight common points and divergences for the reader to understand each method easily, with a focus on key methods (see Table 1)

1.1 Notations

Throughout the paper, methods will explain a machine learning model $h : \mathcal{X} \rightarrow \mathcal{Y}$, with $\mathcal{X}$ and $\mathcal{Y}$ being respectively the input and output domain. Especially, this model is parameterized by the weights $\theta \in \Theta \subseteq \mathbb{R}^d$. If not specified otherwise, h is trained on a training dataset $\mathcal{D}_{train} \subset (\mathcal{X} \times \mathcal{Y})$ of size n with the help of a loss function $l : (\mathcal{X}, \mathcal{Y}, \Theta) \rightarrow \mathbb{R}$. We denote a sample by the tuple $z = (x, y) | \quad x \in \mathcal{X}, y \in \mathcal{Y}$. When an index subscript as i or j is added, *e.g.* z_i, it is assumed that z_i belongs to the training dataset. If the subscript "test" is added, z_{test}, the sample does not belong to the training data. When there is no subscript, the sample can either be or not in the training data. Finally, the empirical risk function is denoted as $\mathcal{L}(\theta) := \frac{1}{n} \sum_{(x,y) \in \mathcal{D}_{train}} l(x, y, \theta) = \frac{1}{n} \sum_{z_j \in \mathcal{D}_{train}} l(z_j, \theta)$, the parameters that minimized this empirical risk as $\theta^* := \arg\min_\theta \mathcal{L}(\theta)$ and an estimator of θ^* is denoted $\hat{\theta}$.

2 Similar Examples

In the XAI literature, similar examples, also referred to as factuals (see Fig. 2), are often used as a way to provide intuitive and interpretable explanations. The core idea is to retrieve the most similar, or the closest, elements in the training set to a sample under investigation z_{test} and to use them as a way to explain a model's output. Specifically, Case-Based Reasoning (CBR) is of particular interest as it mimics the way humans draw upon past experiences to navigate novel situations [38,100]. For example, when learning to play a new video game, individuals do not typically begin from a complete novice level. Instead, they rely on their pre-existing knowledge and skills in manipulating game controllers and draw upon past experiences with similar video games to adapt and apply strategies that have been successful in the past. As described

by Aamodt and Plaza [1], a typical CBR cycle can be delineated by four fundamental procedures: i) RETRIEVE: Searching for the most analogous case or cases, ii) REUSE: Employing the information and expertise extracted from that case to address the problem, iii) REVISE: Modifying the proposed solution as necessary, iv) RETAIN: Preserving the pertinent aspects of this encounter that could be beneficial for future problem-solving endeavors. In addition to being intuitive, the cases retrieved by a CBR system for a given prediction are natural explanations for this output.

While CBR systems are a must-know in the XAI literature, we will not review them as they have already been well analyzed, reviewed, motivated, and described many times [26,28,102]. Instead, the focus here is on case-based explanations (CBE) [102]. CBE are methods that use CBR to explain other systems, also referred to as twin systems [57,60]. In particular, explanations of the system under inspection are generally the outcomes of the RETRIEVE functionality of the twinned CBR system, which oftentimes relies on k-nearest neighbor (k-NN) retrieval [24]. The idea behind k-NN is to retrieve the k most similar training samples (cases) to a test sample z_{test}.

2.1 Factual Methods

One of the main challenges with CBE methods is to define similarity. Indeed, there are many ways of defining similarity measures, and different approaches are appropriate for different representations of a training sample [28]. Generally, CBR systems assume that similar input features are likely to produce similar outcomes. Thus, using a distance metric defined on those input features engenders a similarity measure: the closer the more similar they are. One of the simplest is the unweighted Euclidean distance:

$$dist(z, z') = ||x - x'||_2 \quad | \quad z = (x, y) \in (\mathcal{X} \times \mathcal{Y}) \tag{1}$$

However, **where** – *i.e.* in which space – the distance is computed does have major implications. As pointed out by Hanawa *et al.* [46], the input space does not seem to bring pieces of information on the internal working of the model under inspection but provides more of a data-centric analysis. Thus, recent methods rely instead on either computing the distance in a latent space or weighting features for the k-NN algorithm [31].

Computing distance in a latent space is one possibility to include the model in the similarity measure which is of utmost importance if we want to explain it, as pointed out by Caruana *et al.* [20]. Consequently, they suggested applying the Euclidean distance on the last hidden units h_{-1} of a trained Deep Neural Network (DNN) as a similarity that considers the model's predictions:

$$dist_{DNN}(z, z') = ||h_{-1}(x) - h_{-1}(x')||_2 \quad | \quad z = (x, y) \in (\mathcal{X} \times \mathcal{Y}) \tag{2}$$

Similarly, for convolutional DNN, Papernot and McDaniel [92], and Sani *et al.* [98] suggested conducting the k-NN search in the latent representation of the network and using the cosine similarity distance.

Weighting features is another popular paradigm in CBE. For instance, Shin *et al.* [106] proposed various **global weighting** schemes – *i.e.* methods in which the weights assigned to each input's feature remain constant across all samples as in Eq. (3) – where the weights are computed using the trained network to reveal the input features that were the most relevant for the network's prediction.

$$dist_{features\_weights}(z, z') = ||w(\hat{\theta})^T(x - x')||_2 \quad | \quad z = (x, y) \in (\mathcal{X} \times \mathcal{Y}) \tag{3}$$

Alternatively, Park *et al.* [93] examined **local weighting** by considering varying feature weights across the instance space. However, their approach is not *post-hoc* for DNN. Besides, Nugent *et al.* [89] also focused on local weighting and proposed a method that can be applied to any black-box model. However, their method involves generating multiple synthetic datasets around a specific sample, which may not be suitable for explaining a large number of samples or high-dimensional inputs. In the same line of work, Kenny and Keane [60,61] proposed COLE, by suggesting the direct k-NN search in the attribution space – *i.e* computing saliency maps [7,107,110] for all instances and performing a k-NN search in the resulting dataset of attributions. By denoting $c(\hat{\theta}, z)$ the attribution map of the sample z for the model parameterized by $\hat{\theta}$ gives:

$$dist_{COLE}(z, z') = ||c(\hat{\theta}, z) - c(\hat{\theta}, z')||_2 \tag{4}$$

They used three saliency map techniques [7,107,110] but nothing prevents one to leverage any other saliency map techniques. However, we should also point out that Fel *et al.* [34] questioned attribution methods' ability to truly capture the internal process of DNN. Additionally in [61], Kenny and Keane proposed to use the Hadamard product of the gradient times the input features as a contribution score in the case of DNN with non-linear outputs.

2.2 Conclusions on Similar Examples

Presenting similar examples to an end-user as an explanation for a model's outcomes has been shown through user studies [53,114] and psychology [32] to be generally more convincing than other approaches. However, the current limitations of similarity-based XAI are still significant. For instance, computing a relevant distance between z_{test} and every training data point becomes computationally prohibitive for large datasets. Thankfully, there are efficient search techniques available, as mentioned in the paper by Bhatia *et al.* [14].

Furthermore, **where** the distance is computed does have major implications [46]. Consequently, authors have suggested different feature spaces or weighting schemes to investigate, but their relevance to reflect the inner workings of a model remains questionable. In addition, it is still unclear in the literature if one approach prevails over others. In this regard, it is relevant to point out that psychological studies [32,78,88,112] underscore the importance of shared features, overall resemblance, context, and the interplay between perceptual and conceptual factors in similarity judgments. In fact, we can point out that none of the current factual methods leverage all those aspects at once.

Finally, considering the position of retrieved similar examples in relation to a model's decision boundaries is crucial for relevant explanations. Neglecting this can confuse users if factual examples contradict the model's prediction. Contrastive explanations address this issue and are discussed in Sect. 3.

3 Contrastive Explanations

Contrastive explanations are a class of explanation that provides the consequences of another plausible reality, the repercussion of changes in the model's input [17,113]. More simply, they are explanations where we modify the input and observe the reaction of the model's prediction, the modified input is returned as the explanation and its meaning depends on the model's prediction of it. Those methods are mainly *post-hoc* methods applied to classification models. This includes i) counterfactuals (CF): *an imagined alternative to reality about the past, sometimes expressed as "if only ..." or "what if ..."* [17], ii) semi-factuals (SF): *an imagined alternative that results in the same outcome as reality, sometimes expressed as "even if ..."* [17], and iii) adversarial examples (perturbations or attacks) (AP): *inputs formed by applying small but intentionally worst-case perturbations to examples from the dataset, such that the perturbed input results in the model outputting an incorrect answer with high confidence* [41]. Examples of those three formats are provided in Fig. 2 from Kenny and Keane [62].

AP and CF are both perturbations with an expected change in the prediction, they only differ in the goal as CF attempt to provide an explanation of the model's decision while AP are mainly used to evaluate robustness. In fact, AP can be considered CF [115], and for robust models, AP methods can generate interpretable CF [105]. Nonetheless, AP are hardly perceptible perturbations designed to fool the model [111], therefore, they are generative and those methods will not be further detailed in this work. Then, we can generalize SF and CF, with a given distance $dist$, and the examples conditioned space $\mathcal{X}_{cond(f,x)} \subset \mathcal{X}$:

$$CF(x_{test}) := \underset{x \in \mathcal{X}_{cond(f,x_{test})} | h(x) \neq h(x_{test})}{\arg\min} dist(x_{test}, x) \tag{5}$$

$$SF(x_{test}) := \underset{x \in \mathcal{X}_{cond(f,x_{test})} | h(x) = h(x_{test})}{\arg\max} dist(x_{test}, x) \tag{6}$$

For natural CF and SF, the input space is conditioned to the training set, $\mathcal{X}_{cond(f,x_{test})} = X_{train}$. While for AP, there is no condition on the input space, in Eq. (5), $\mathcal{X}_{cond(f,x_{test})} = \mathcal{X}$. The distance and the condition of the input space are the key differences between CF and SF methods.

This section discusses both counterfactuals and semi-factuals as they are often treated together in the literature [17,25,42,62]. The literature for both formats is large in social sciences and in XAI for generative methods, hence we will extract key findings before presenting natural example-based methods.

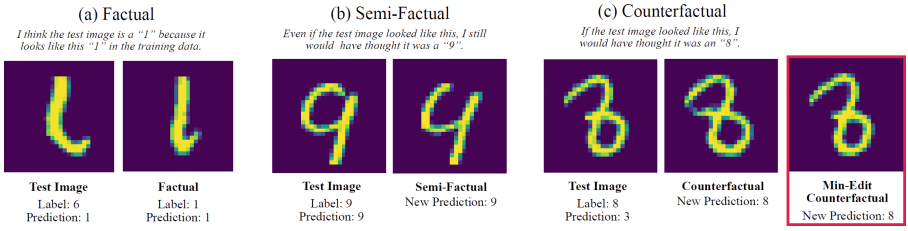

Fig. 2. Illustration of factuals, SF, and CF from Kenny and Keane [62]. The factual makes us understand the misclassification, while SF and CF show us how far or close the decision boundary is. Min-edit represents the AP, as differences are not visible.

3.1 Counterfactuals

The social science grounding of counterfactuals is deep, either in philosophy, or psychology. Indeed, the search for CF's semantic definition goes back a long time [13,44,72], and historically revolves around the notion of cause and effect, sometimes called facts and foils [75,79]. Then, Halpern and Pearl [44] argued that providing the cause of an event answers the question "Why?" and thus, provides a powerful explanation. Moreover, the philosophical literature argued that CF allow us to communicate and understand the causal relation between facts and foils [72,79]. Psychology also possesses a rich literature regarding CF [17,97], which has continued to evolve in recent years [18,59,80] thanks to the arrival of CF in XAI through Wachter *et al.* [115]. Humans' natural use of counterfactuals in many situations was highlighted by Byrne [17]: *From amusing fantasy to logical support, they explain the past, prepare the future, modulate emotional experience, and support moral judgments.* Furthermore, when people encounter CF they have both the counterfactual and the factual in mind [18]. The insights from philosophy and psychology [18,80] have shown the pertinence and potential of CF as well as SF for XAI. To match such promises, CF in XAI need to verify the definitions and properties of CF typically employed by humans.

Expected properties for natural CF can be extrapolated from conclusions and discovered properties in XAI for generated CF even though the literature on natural CF is slim. Such desirable properties for CF, derived from social sciences, could be summarized as follows: i) **plausibility** [58,59,113]: CF should be as realistic as possible; ii) **validity** [84]: if the model's prediction on CF differ from the prediction on the query (see the definition (5)); iii) **sparsity** [58, 84,113]: the number of features that were changed between CF and the query should be as little as possible; iv) **diversity** [54,84]: if several CF are proposed, they should be different from each other; v) **actionability** [58,113]: the method should allow the user to select features, to modify and specify immutable ones; vi) **proximity** [54,58,59,84]: CF should be as close as possible to the query.

Counterfactuals Methods: Keane *et al.* [59] argued that nearest unlike neighbors (NUN) [27] a derivative of nearest neighbors [24], are the ancestors of counterfactuals in XAI. NUN are the nearest element to the query that belongs to a different class. They are natural CF when the class is given by the model prediction. Natural counterfactuals and semi-factuals are faced with the same discussions around similarity as factuals section. However, here, the similarity should take into account sparsity.

NUN were first used in XAI by Doyle *et al.* [29,90] but not as an explanation, only to find SF. The only method to the best of our knowledge that uses NUN as explanations is KLEOR from Cummins and Bridge [25], they provided it as a complement to SF explanations to give intuition on the decision boundary. Nonetheless, they highlighted that the decision boundary might be much more complex than what the SF and CF pairs can reveal. Indeed, a line between SF and CF may intersect the decision boundary several times, which can lead to explanations that are not always faithful. Furthermore, Keane *et al.* [59] argued that "good natural counterfactuals are hard to find" as the dataset's low density may prevent sparse and proximal natural CF.

Counterfactuals as known in XAI were introduced by Wachter et al. [115], and flourished through generative methods as shown by the numerous surveys [54,84,113]. Two periods emerge: one focused on interpretable tabular data [113], and the other on complex data like images [6,62]. While generating plausible instances for the first period was not an issue it remains challenging for the second, even with diffusion models [6]. More research is needed to explore natural counterfactuals with their inherent plausibility [59,113]. Moreover, adversarial perturbations proved that for non-robust DNN, a generated example close to a natural instance is not necessarily plausible.

To conclude on counterfactuals, their large literature produced expected properties with deep social science grounding. Such desiderata highlight the pros and cons between generative and natural CF. Indeed, for high dimensional data, the reader is faced with the choice of simple and plausible natural CF or proximal and sparse generated CF through a model explaining another model.

3.2 Semi-factuals

SF literature is most of the time included in the CF literature be it in philosophy [42], psychology [17], or XAI [25,62]. In fact, SF, "even if ..." are semantically close to CF, "what if ..." [5,13,42], (see Eqs. (5) and (6)). However, psychology has demonstrated that human reactions differ between CF and SF. While CF strengthen the causal link between two elements, SF reduce it [18], CF increase fault and blame in a moral judgment while SF diminish it.

Expected properties for CF and SF were inspired by social science, hence, because of their close semantic definition, many properties are common between both: SF should also respect their definition in Eq. (6) (**validity**), then to

make the comparison possible and relevant they should aim towards **plausibility** [5], **sparsity** [5], **diversity**, and **actionability**. Nonetheless, the psychological impact of CF and SF differ, hence there are also SF properties that contrast with CF properties. The difference between equations (5) and (6) – *i.e.* arg min vs arg max – suggests that to replace CF's proximity, SF should be the farthest from the studied sample, while not crossing the decision boundary [25]. As such, we propose the **decision boundary closeness** as a necessary property, and a metric to evaluate it could be the distance between SF and SF's NUN. Finally, SF should not go in any direction from the studied sample but aim toward the closest decision boundary. Therefore, it should be aligned with NUN [25,29,90], this property was not named, we suggest calling it **counterfactual alignment**.

Semi-factuals methods were first reviewed in XAI by a recent survey from Aryal and Keane [5]. They divided SF methods and history into four parts. The first three categories consist of one known method that will illustrate them:

- **SF based on feature-utility**, Doyle *et al.* [29] discovered that similar examples may not be the best explanations and suggested giving examples farther from the studied sample. To find the best explanation case, *dist* in Eq. (6) is a utility evaluation based on features difference.
- **NUN-related SF**, Cummins and Bridge [25] proposed KLEOR where Eq. (6)'s *dist* is based on NUN similarity. Then, they penalize this distance to make sure the SF are between the query and nearest unlike neighbors.
- **SF near local-region boundaries**, Nugent *et al.* [90] approximate the decision boundary of the model in the neighborhood of the studied sample through input perturbations (like LIME [96]). Then SF are given by the points that are the closest to the decision boundary.
- **The modern era: *post*-2020 methods**, inspired by CF methods, many generative methods emerged in recent years [55,62].

To conclude, semi-factuals are a natural evolution of factuals. Moreover, their complementarity with counterfactuals was exposed through the literature, first to find and evaluate SF, then to provide a range to the decision boundary. Finally, generative and natural SF possess the same pros and cons as CF ones.

Even though contrastive explanations bring insights into a model's behavior, it has no impact on the current model situation, what led to this state, or how to change it. Contrastively, influential instances (see Sect. 4) extract the samples with the most influence on the model's training. Removing such samples from the training set will have a huge impact on the resulting model.

4 Influential Examples

Influential instances could be defined as instances more likely to change a model's outcome if they were not in the training dataset. Furthermore, such measures of influence provide one with information on "in which direction" the model

decision would have been affected if that point was removed. Being able to trace back to the most influential training samples for a given test sample z_{test} has been a topic of interest mainly for example-based XAI.

4.1 Influential Instances Methods

Influence functions originated from robust statistics in the early 70 s. In essence, they evaluate the change of a model's parameters as we up-weight a training sample by an infinitesimal amount [45]: $\hat{\theta}_{\epsilon,z_j} := \arg\min_\theta \mathcal{L}(\theta) + \epsilon l(z_j, \theta)$. One way to estimate the change in a model's parameters of a single training sample would be to perform *Leave-One-Out* (LOO) retraining, that is, to train the model again with the sample of interest being held out of the training dataset. However, repeatedly re-training the model to exactly retrieve the parameters' changes could be computationally prohibitive, especially when the dataset size and/or the number of parameters grows. As removing a sample z_j can be linearly approximated by up-weighting it by $\epsilon = -\frac{1}{n}$, computing influence helps to estimate the change of a model's parameters if a specific training point was removed. Thus, by making the assumption that the empirical risk $\mathcal{L}$ is twice-differentiable and strictly convex with respect to the model's parameters θ making the Hessian $H_{\hat{\theta}} := \frac{1}{n} \sum_{z_i \in \mathcal{D}_{train}} \nabla^2_\theta l(z_i, \hat{\theta})$ positive definite, Cook and Weisberg [23] proposed to compute the influence of z_j on the parameters $\hat{\theta}$ as:

$$\mathcal{I}(z_j) := -H^{-1}_{\hat{\theta}} \nabla_\theta l(z_j, \hat{\theta}) \tag{7}$$

Later, Koh and Liang [68] popularized influence functions in the machine learning community as they took advantage of auto-differentiation frameworks to efficiently compute the hessian for DNN and derived Eq. (7) to formulate the influence of up-weighting a training sample z_j on the loss at a test point z_{test}:

$$\text{IF}(z_j, z_{test}) := -\nabla_\theta l(z_{test}, \hat{\theta})^T H^{-1}_{\hat{\theta}} \nabla_\theta l(z_j, \hat{\theta}) \tag{8}$$

This formulation opens its way into example-based XAI as it compares to the study of finding the nearest neighbors of z_{test} in the training dataset – *i.e.* the most similar examples (Sect. 2) – with two major differences though: i) points with high training loss are given more influence *revealing that outliers can dominate the model parameters* [68], and ii) $H^{-1}_{\hat{\theta}}$ measures what Koh and Liang called: *the resistance of the other training points to the removal of* z_j [68]. However, it should be noted that hessian computation remains a significant challenge, that could be alleviated with common techniques [3,77,101]. By normalizing Eq. (8), Barshan *et al.* [10] further added stability to the formulation.

Oftentimes, we are not only interested in individual instance influence but in the influence of a group of training samples (*e.g.* mini-batch effect, multi-source data, etc.). Koh *et al.* [69] suggested that using the sum of individual influences as the influence of the group constitutes a good proxy to rank those groups in terms of influence. Basu *et al.* [12] on their side suggested using a second-order approximation to capture possible cross-correlations but they specified it is most

likely impracticable for DNN. In a later work, Basu *et al.* [11] concluded that influence function estimates for DNN are fragile as the assumptions on which they rely, being near optimality and convexity, do not hold in general for DNN.

LOO approximation is one of the previously mentioned motivations behind influence estimates as it avoids the prohibitive LOO retraining required for every sample in the training data. Thus, some authors proposed approaches that optimize the number of LOO retraining necessary to get a grasp on a sample's influence such as Feldman and Zhang [36]. Although this significantly reduces the number of retraining compared to naive LOO retraining, it still requires a significant amount of them. Recently, a new approach that relates to influence functions and involves training many models, was introduced with data models [52,99] which we do not review here.

As Basu *et al.* [11] pointed out, there is a discrepancy between LOO approximation and influence function estimates, especially for DNN. However, Bae *et al.* [9] claimed that this discrepancy is due to influence functions approaching what they call the proximal Bregman response function (PBRF), rather than approximating the LOO retraining, which does not interfere with their ability to perform the task they were thought for, especially XAI. Thus, they suggested evaluating the quality of influence estimates by comparing them to the PBRF rather than LOO retraining as it was done until now.

Influence computation that relies on kernels is another paradigm to find the training examples that are the most responsible for a given set of predictions. For instance, Khanna *et al.* [63] proposed an approach that relies on Fisher's kernels and they related it to the one from Koh and Liang [68] as a generalization of the latter under certain assumptions. Yeh *et al.* [117] also suggested an approach that leverages kernels but this time they relied on the representer theorem [103]. That allows them to focus on explaining only the *pre-activation prediction layer* of a DNN for classification tasks. In addition, their influence scores, called representer values, provide supplementary information, with positive representer values being excitatory and negative values being inhibitory. However, this approach requires introducing an $L2$ regularizer during optimization, which can prevent *post-hoc* analysis if not responsible for training. Additionally, Sui *et al.* [109] argued that this approach provides more of a *class-level* explanation rather than an *instance-level* explanation. To address this issue and the $L2$ regularizer problem, they proposed a method that involves hessian computation on the classification layer, with only the associated computational cost. However, the ability to retrieve relevant samples when investigating only the final prediction layer was questioned by Feldmann and Zhang [36], who found that memorization does not occur in the last layer.

Tracing the training process has been another research field to compute influence scores. It relies on the possibility to replay the training process by

saving some checkpoints of our model parameters, or states, and reloading them in a post-hoc fashion [22,47,95]. In contrast to the previous approaches, they rely neither on being near optimality nor being strongly convex, which is more realistic when we consider the reality of DNN. However, they require handling the training procedure to save the different checkpoints, potentially numerous, hence they are intrinsic methods, which in practice is not always feasible.

4.2 Conclusions on Influential Instances

Influential techniques can provide both global and local explanations to enhance model performance. Global explanations allow for the identification of training samples that significantly shape decision boundaries or outliers (see Fig. 1), aiding in data curation. On the other hand, local explanations offer guidance for altering the model in a desired way (see Fig. 3). Although they have been compared to similar examples and have been shown to be more relevant to the model [46], they are more challenging to interpret and their effectiveness for trustworthiness is unclear. Further research, particularly user studies, is necessary to determine their ability to take advantage of human cognitive processes.

Fig. 3. Figure taken from F. Liu [95]: A tracing process for estimating influence, TracIn, applied on ImageNet. The first column is composed of the test sample, the next three columns display the training examples that have the most positive value of influence score while the last three columns point out the training examples with the most negative values of influence score. (fr-bulldog: french-bulldog)

5 Prototypes

Prototypes are a set of representative data instances from the dataset, while criticisms are data instances that are not well represented by those prototypes [64]. Figure 4 shows examples of prototypes and criticisms from Imagenet dataset.

Fig. 4. Figure taken from [64]: Learned prototypes and criticisms from Imagenet dataset (two types of dog breeds)

5.1 Prototype Methods

Prototypes and criticism can be used to add data-centric interpretability, *post-hoc* interpretability, or to build an interpretable model [83]. The data-centric approaches will be briefly introduced.

Prototypes for Data-Centric Interpretability: Clustering algorithms that return actual data points as cluster centers such as k-medoids methods [56,87] could be used to better understand the data distribution. We can consider the cluster centers as prototypes.

The abundance of large datasets has renewed the interest in the data summarization methods [8,73,74,82,108], which consist of finding a small subset of data points that covers a large dataset. The subset elements can be considered prototypes. Additionally, we found data summarization methods based on the Maximum Mean Discrepancy (MMD), such as MMD-critic [64] and Protodash [43], that learn both prototypes and criticisms.

Prototypes for *Post-hoc* Interpretability: Most prototype methods are data-centric that provide no information on the model. However, such methods can be computed in a meaningful search space for the model as done with similar examples Sect. 2.1 and give global explanations with the model vision of the dataset. Similarly, local explanations can be extracted by comparing studied samples to the closest prototypes in the search space. But to our knowledge, only one method explores such a possibility. Filho *et al.* [37] proposed M-PEER (Multiobjective Prototype-based Explanation for Regression) method that finds the prototypes using both the training data and the model output. It optimizes the error of the explainable model and the fidelity and interpretability metrics.

Prototype-Based Models Interpretable by Design: After data-centric and *post-hoc* methods, there are methods that construct prototype-based models. Those models are interpretable by design because they provide a set of prototypes that make sense for the model, those methods are mainly designed for classification. An interpretable classifier learns a set of prototypes $P_c \subseteq \{(x, y) \in$

$\mathcal{D}_{train}|y = c\}$. Each P_c captures the full variability of class c while avoiding confusion with other classes. The learned prototypes are then used by the model to classify the input. We identified three types of prototype-based classifiers:

- **Classifiers resolving set cover problems** select convex sets that cover each class with prototypes to represent it. Various types of convex sets such as boxes, balls, convex hulls, and ellipsoids can be used. Class Cover Catch Digraphs (CCCD) [76] and ProtoSelect [15] used balls where the centers were considered prototypes. Then, the nearest-prototype rule is used to classify the data points. CCCD finds, for each class c, a variable number of balls that cover all points of class c and no points of other classes. Its radius is chosen as large as possible. However, even within large classes, there can still be a lot of interesting within-class variability that should be taken into account when selecting the prototypes. To overcome this limitation, ProtoSelect used a fixed radius across all points, to allow the selection of multiple prototypes for large classes, and they also allow wrongly covered and non-covered points. They simultaneously minimize three elements: i) the number of prototypes; ii) the number of uncovered points; iii) the number of wrongly covered points.
- **Classifiers using Bayesian models for explanation**, Kim *et al.* [65] proposed the Bayesian Case Model (BCM) that extends Latent Dirichlet Allocation [16]. In BCM, the idea is to divide the data into s clusters. For each cluster, a prototype is defined as the sample that maximizes the subspace indicators that characterize the cluster. When a sample is given to BCM, this last one yield a vector of probability to belong to each of the s clusters which can be used for classification. Thus, the classifier uses as an input a vector of dimension s, which allows the use of simpler models due to dimensionality reduction. In addition, the prototype of the most likely cluster can then be used as an explanation.
- **Classifiers based on neural networks** learn to select prototypes defined in the latent space, which are used for the classification. This lead to a model that is more interpretable than a standard neural network since the reasoning process behind each prediction is "transparent". Learning Vector Quantization (LVQ) [70] is widely used for generating prototypes as weights in a neural network. However, the use of generated prototypes reduces their interpretability. ProtoPNet [21] also stocks prototypes as weights and trains them, but projects them to training samples patches representation during training. Given an input image, its patches are compared to each prototype, the resulting similarity scores are then multiplied by the learned class connections of each prototype. ProtoPNet has been extended to time series data via ProSeNet [81], or with a more interpretable structure with ProtoTree [86] and HPNet [48]. Instead of using linear bag-of-prototypes, ProtoTree and HPNet used hierarchically organized prototypes to classify images. ProtoTree improves upon ProtoPNet by using a decision tree which provides an easy-to-interpret global explanation and can be used to locally explain a single prediction. Each node in this tree contains a prototype (as defined by ProtoPNet) and the similarity scores between image patches and the prototypes

are used to determine the routing through the tree. Decision-making is therefore similar to human reasoning [86]. Nauta *et al.* [85] proposed a method called "This Looks Like That, Because" to understand prototypes similarities. This method allows checking why the model considered two examples as similar. For instance, it is possible that a human thinks that the common point between two examples is their color, while the model uses their shape. The method modifies some characteristics of the input image, such as hue, or shape, to observe how the similarity score changes. This allows us to measure the importance of each of these characteristics.

5.2 Conclusions on Prototypes

Most prototype methods are data-centric, but we have seen that applying such methods in a meaningful space for the model can bring *post-hoc* global and local explanations. Nonetheless, a second part of the literature constructs prototype-based classifiers explainable by design, those methods are promising and produce models with natural reasoning but adapting a new model to such architecture can be prohibitive.

6 Concept-Based XAI

Prototype-based models compare prototypical parts, *e.g.* patches, and the studied sample to make the classification. The idea of parts is not new to the literature, the part-based explanation field, developed for fine-grained classification, is able to detect semantically significant parts of images. The first part-based model required labeled parts for training and can be considered object detection with a semantic link between the detected objects. Afterward, unsupervised methods such as OPAM [94] or Particul [116] emerged, those methods still learned classification in a supervised fashion, but no labels were necessary for part identification. In fact, the explanation provided by this kind of method can be assimilated into concept-based explanations. A concept is an abstraction of common elements between samples, as an example Fig. 5 shows the visualization of six different concepts that the CRAFT method [35] associated with the given image. To understand parts or concepts, the method uses examples and supposes that with a few examples, humans are able to identify the concept.

6.1 Concepts Methods

Like in part-based XAI, the first concept-based method used labeled concepts. Kim et al. [66] introduced concept activation vectors (CAV) to represent concepts using a model latent space representation of images. Then, they design a post-hoc method, TCAV [66] based on CAV to evaluate an image correspondence to a given concept. Even though it seems promising, this method requires prior knowledge of the relevant concepts, along with a labeled dataset of the associated concepts, which is costly and prone to human biases.

Fig. 5. Illustration from Fel *et al.* [35]. Natural examples in the colored boxes define a concept. **Purple box**: could define the concept of "**chainsaw**". **Blue box**: could define the concept of "**saw's motor**". **Red box**: could define the concept of "**jeans**". (Color figure online)

Fortunately, recent works have been conducted to automate the concept discovery in the training dataset without humans in the loop. For instance, ACE, proposed by Ghobarni et al. [39], employs a semantic segmentation technique on images belonging to a specific class of interest and use an Inception-V3 neural network to compute activations of an intermediate model layer for these segments. The resulting activations are then clustered to form a set of prototypes, which they refer to as "concepts". However, the presence of background segments in these concepts requires a post-processing clean-up step to remove irrelevant and outlier concepts. Zhang et al. [118] proposed an alternative approach to solving the unsupervised concept discovery problem through matrix factorizations [71] in the networks' latent spaces. However, such methods operate at the convolutional kernel level, which may lead to concepts based on shape and/or ignore more abstract concepts.

As an answer, Fel et al. [35] proposed CRAFT, which uses Non-Negative Matrix Factorization [71] for concept discovery. In addition to filling in the blank of previous approaches, their method provides an explicit link between the concepts' global and local explanations (Fig. 5). While their approach alleviates the previously mentioned issues, the retrieved concepts are not always interpretable. Nonetheless, their user study proved the pertinence of the method.

6.2 Conclusions on Concepts

Concept-based explanations allow *post-hoc* global and local explanations, by understanding the general concepts associated with a given class and the concepts used for a decision. We draw attention to methods that do not require expert knowledge to find out relevant concepts as they are prone to human bias. Even though automated concept discovery is making tremendous progress, the interpretation of such concepts and their ability to gain users' trust stay questionable as very few user studies have been conducted on the subject.

Table 1. Comparison table between the different natural example-based formats and methods. NA: Not applicable, FGCV: Fine-grained computer vision

SIMILAR EXAMPLES	Year	Global/Local	Post-hoc/Intrinsic	Model or data-type specificity	Distance	Weighting
Caruana et al. [20]	1999	Local	Post-hoc	DNN	Euclidean	None
Shin et al. [106]	2000	Local	Post-hoc	DNN	Euclidean	Global
Park et al. [93]	2004	Local	Intrinsic	DNN	Euclidean	Local
Nugent et al. [89]	2005	Local	Post-hoc	None	Euclidean	Local
Sani et al. [98]	2017	Local	Post-hoc	Deep CNN	Cosine similarity	Local
Papernot and McDaniel [92]	2018	Local	Post-hoc	Deep CNN	Cosine similarity	Local
Cole [60] [61]	2019	Local	Post-hoc	None	Euclidean	Local with attributions
CONTRASTIVE EXPLANATIONS	Year	Global/Local	Post-hoc/Intrinsic	Model or data-type specificity	Semi-factual group of method	
Doyle et al. [29,30]	2004	Local	Post-hoc	None	SF based on feature-utility	
NUN [25,27,29]	2006	Local	Post-hoc	None	Natural CF	
KLEOR [25]	2006	Local	Post-hoc	None	NUN-related SF	
Nugent et al. [90]	2009	Local	Post-hoc	None	Local-region boundaries	
INFLUENTIAL INSTANCES	Year	Global/Local	Post-hoc/Intrinsic	Model or data-type specificity		Requires model's gradients
Koh and Liang [68]	2017	Both	Post-hoc	$\mathcal{L}$ twice-differentiable and strictly convex w.r.t. θ		Yes
Khanna and al. [63]	2018	Local	Post-hoc	Requires an access to the function and gradient-oracles		Yes
Yeh and al. [117]	2018	Local	Intrinsic	Work for classification neural networks with regularization		Yes
Hara and al. [47]	2019	Local	Intrinsic	Models trained with SGD, saving intermediate checkpoints		Yes
Koh and Liang [69]	2019	Both	Post-hoc	$\mathcal{L}$ twice-differentiable and strictly convex w.r.t. θ		Yes
Basu and al. [12]	2019	Both	Post-hoc	$\mathcal{L}$ twice-differentiable and strictly convex w.r.t. θ		Yes
Barshan and al. [10]	2020	Both	Post-hoc	$\mathcal{L}$ twice-differentiable and strictly convex w.r.t. θ		Yes
Feldman and Zhang [36]	2020	Global	Intrinsic	Requires to train numerous models on subsampled datasets		No
Pruthi and al. [95]	2020	Local	Intrinsic	Requires saving intermediate checkpoints		Yes
Sui and al. [109]	2021	Local	Post-hoc	Work for classification neural networks		Yes
Chan and al. [22]	2021	Both	Intrinsic	Requires saving intermediate checkpoints		Yes
PROTOTYPES	Year	Global/Local	Post-hoc/Intrinsic	Model or data-type specificity	Task	Other
CCCD [76]	2003	Both	NA	by-design	Classification	Set cover
ProtoSelect [15]	2011	Both	NA	by-design	Classification	Set cover
Kim et al. [65]	2019	Both	NA	by-design, tabular	Classification	Bayesian-based
ProtoPNet [21]	2019	Both	NA	by-design, FGCV	Classification	Neural network
ProSeNet [81]	2019	Both	NA	by-design, sequences	Classification	Neural network
ProtoTree [86]	2021	Both	NA	by-design, FGCV	Classification	Neural network
M-PEER [37]	2023	Both	Post-hoc	No	Regression	NA
CONCEPTS	Year	Global/Local	Post-hoc/Intrinsic	Model or data-type specificity	Need labeled concepts	Concepts format
OPAM [94]	2017	Global	NA	By-design, FGCV	Yes	part-based
TCAV [66]	2018	Global	Post-hoc	Neural network	Yes	same as input
ACE [39]	2019	Global	Post-hoc	Neural network	No	segmented parts
Zhang et al. [118]	2021	Global	Post-hoc	Neural network	No	segmented parts
CRAFT [35]	2022	Global	Post-hoc	Neural network	No	crops
Particul [116]	2017	Global	NA	By-design, FGCV	Yes	part-based

7 Conclusions and Discussions

This paper explored explainability literature about natural example-based explainability and provided a general social science justification for example-based XAI. We described each kind of explanation possible through samples. For each possibility, we reviewed what explanation they bring, then classified and presented the major methods. We summarize all explored methods in Table 1. We saw that all those methods are based on a notion of similarity. As such, for them to explain the model, the similarity between instances should take into account the model. There are two ways of doing it: project the instances in a meaningful space for the model and/or weight instances. Hence, similarity definitions from factuals (Sect. 2.1) can be ported to other formats and social science groundings could also be shared. However, if the training data is sparse in the search space, finding cases with good properties for a given format may be challenging.

Among the formats, contrastive explanations, prototypes, and concept examples can be generated, which brings competition to non-generative methods. We argue that both generative and natural examples have their pros and cons.

Indeed, natural examples are simple to compute and ensure plausibility while generated examples can be more proximal and sparse but require a model to explain another model (see Sect. 3.1 for properties definitions).

We have illustrated that the different example-based formats bring different kinds of explanations, and each one has its own advantages, Fig. 1 shows their diversity, have their scope of application, and complementarity. To summarize those advantages non-exhaustively: i) Factuals give confidence in the decisions of the model and are pertinent in AI-assisted decisions. ii) For classification, contrastive explanations give local insight into the decision boundary. iii) Influential instances explain how samples influenced the model training. iv) Prototypes and concepts give information on the whole model behavior, but may also be used to explain decisions. Nonetheless, like all explanations, we cannot be sure that humans will have a correct understanding of the model or the decision. Furthermore, there is no consensus on how to ensure a given method indeed explains the decisions or inner workings of the model. Moreover, for example-based explainability, the data is used as an explanation, hence, without profound knowledge of the dataset, humans will not be able to draw conclusions through such explanations. Therefore, the evaluation of example-based methods should always include a user study, which are scarce in this field and in XAI in general, especially with the lack of availability and consensus around quantitative metrics to evaluate example-based explanations. Finally, we hope our work will motivate, facilitate and help researchers to keep on developing the field of XAI and in particular, natural example-based XAI and to address the identified challenges.

Acknowledgments. This work has been supported by the French government under the "France 2030" program as part of the SystemX Technological Research Institute. This work was conducted as part of the Confiance.AI program, which aims to develop innovative solutions for enhancing the reliability and trustworthiness of AI-based systems. Additional funding was provided by ANR-3IA Artificial and Natural Intelligence Toulouse Institute (ANR-19-PI3A-0004).

We are also thankful to the DEEL's (https://www.deel.ai/) core team for their expertise and feedbacks. A.M. Picard, D. Vigouroux, C. Friedrich, V. Mussot, and Y. Prudent.

Finally, we are thankful to the authors who accepted our use of their figures. E.M Kenny and M.T. Keane [61,62], F. Liu [95], B. Kim [64], and T. Fel [35].

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Commun. **7**, 39–59 (1994)
2. Adadi, A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access **6**, 52138–52160 (2018)
3. Agarwal, N., Bullins, B., Hazan, E.: Second-order stochastic optimization for machine learning in linear time. J. Mach. Learn. Res. **18**, 4148–4187 (2017)
4. Arrieta, A.B., et al.: Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf. Fusion **58**, 82–115 (2020)

5. Aryal, S., Keane, M.T.: Even if explanations: Prior work, desiderata & benchmarks for semi-factual XAI. arXiv preprint arXiv:2301.11970 (2023)
6. Augustin, M., Boreiko, V., Croce, F., Hein, M.: Diffusion visual counterfactual explanations. In: NeurIPS (2022)
7. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS One **10**, e0130140 (2015)
8. Badanidiyuru, A., Mirzasoleiman, B., Karbasi, A., Krause, A.: Streaming submodular maximization: massive data summarization on the fly. In: KDD (2014)
9. Bae, J., Ng, N., Lo, A., Ghassemi, M., Grosse, R.B.: If influence functions are the answer, then what is the question? In: NeurIPS (2022)
10. Barshan, E., Brunet, M.E., Dziugaite, G.K.: RelatIF: identifying explanatory training samples via relative influence. In: AISTATS (2020)
11. Basu, S., Pope, P., Feizi, S.: Influence functions in deep learning are fragile. In: ICLR (2021)
12. Basu, S., You, X., Feizi, S.: On second-order group influence functions for black-box predictions. In: ICML (2020)
13. Bennett, J.: A Philosophical Guide to Conditionals. Clarendon Press (2003)
14. Bhatia, N., et al.: Survey of nearest neighbor techniques. arXiv preprint arXiv:1007.0085 (2010)
15. Bien, J., Tibshirani, R.: Prototype selection for interpretable classification. Ann. Appl. Stat. (2011)
16. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. JMLR **3**, 993–1022 (2003)
17. Byrne, R.M.: Counterfactual thought. Annu. Rev. Psychol. **67**, 135–157 (2016)
18. Byrne, R.M.: Counterfactuals in explainable artificial intelligence (XAI): evidence from human reasoning. In: IJCAI (2019)
19. Cambria, E., Malandri, L., Mercorio, F., Mezzanzanica, M., Nobani, N.: A survey on XAI and natural language explanations. IPM **60**, 103111 (2023)
20. Caruana, R., Kangarloo, H., Dionisio, J., Sinha, U., Johnson, D.: Case-based explanation of non-case-based learning methods. In: AMIA Symposium (1999)
21. Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., Su, J.K.: This looks like that: deep learning for interpretable image recognition. In: NeurIPS (2019)
22. Chen, Y., Li, B., Yu, H., Wu, P., Miao, C.: HYDRA: hypergradient data relevance analysis for interpreting deep neural networks. In: AAAI (2021)
23. Cook, R.D., Weisberg, S.: Residuals and Influence in Regression. Chapman and Hall, New York (1982)
24. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE TIT **13**, 21–27 (1967)
25. Cummins, L., Bridge, D.: Kleor: a knowledge lite approach to explanation oriented retrieval. CAI **25**, 173–193 (2006)
26. Cunningham, P., Doyle, D., Loughrey, J.: An evaluation of the usefulness of case-based explanation. In: Ashley, K.D., Bridge, D.G. (eds.) ICCBR 2003. LNCS (LNAI), vol. 2689, pp. 122–130. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-45006-8_12
27. Dasarathy, B.V.: Nearest unlike neighbor (NUN): an aid to decision confidence estimation. Opt. Eng. **34**, 2785–2792 (1995)
28. De Mantaras, R.L., et al.: Retrieval, reuse, revision and retention in case-based reasoning. KER **20**, 215–240 (2005)

29. Doyle, D., Cunningham, P., Bridge, D., Rahman, Y.: Explanation oriented retrieval. In: Funk, P., González Calero, P.A. (eds.) ECCBR 2004. LNCS (LNAI), vol. 3155, pp. 157–168. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-28631-8_13
30. Doyle, D., Cunningham, P., Walsh, P.: An evaluation of the usefulness of explanation in a case-based reasoning system for decision support in bronchiolitis treatment. Comput. Intell. **22**, 269–281 (2006)
31. Dudani, S.A.: The distance-weighted k-nearest-neighbor rule. IEEE TSMC (1976)
32. Elgin, C.Z.: True Enough. Philosophical Issues (2004)
33. Fel, T., Cadène, R., Chalvidal, M., Cord, M., Vigouroux, D., Serre, T.: Look at the variance! Efficient black-box explanations with Sobol-based sensitivity analysis. In: NeurIPS (2021)
34. Fel, T., et al.: Don't lie to me! robust and efficient explainability with verified perturbation analysis. In: CVPR (2022)
35. Fel, T., et al.: CRAFT: concept recursive activation factorization for explainability. In: CVPR (2022)
36. Feldman, V., Zhang, C.: What neural networks memorize and why: discovering the long tail via influence estimation. In: NeurIPSs (2020)
37. Filho, R.M., Lacerda, A.M., Pappa, G.L.: Explainable regression via prototypes. ACM TELO **2**, 1–26 (2023)
38. Gentner, D.: Structure-mapping: a theoretical framework for analogy. Cogn. Sci. **7**, 155–170 (1983)
39. Ghorbani, A., Wexler, J., Zou, J.Y., Kim, B.: Towards automatic concept-based explanations. In: NeurIPS (2019)
40. Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E.: Peeking inside the black box: visualizing statistical learning with plots of individual conditional expectation. JCGS **24**, 44–65 (2015)
41. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: ICLR (2015)
42. Goodman, N.: The problem of counterfactual conditionals. J. Philos. (1947)
43. Gurumoorthy, K.S., Dhurandhar, A., Cecchi, G., Aggarwal, C.: Efficient data representation by selecting prototypes with importance weights. In: ICDM (2019)
44. Halpern, J.Y., Pearl, J.: Causes and explanations: a structural-model approach. Part II: explanations. BJPS (2005)
45. Hampel, F.R.: The influence curve and its role in robust estimation. JASA **69**, 383–393 (1974)
46. Hanawa, K., Yokoi, S., Hara, S., Inui, K.: Evaluation of similarity-based explanations. In: ICLR (2021)
47. Hara, S., Nitanda, A., Maehara, T.: Data cleansing for models trained with SGD. In: NeurIPS (2019)
48. Hase, P., Chen, C., Li, O., Rudin, C.: Interpretable image recognition with hierarchical prototypes. In: HCOMP (2019)
49. Hastie, T.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, Heidelberg (2009). https://doi.org/10.1007/978-0-387-84858-7
50. Holzinger, A., Saranti, A., Molnar, C., Biecek, P., Samek, W.: Explainable AI methods-a brief overview. In: Holzinger, A., Goebel, R., Fong, R., Moon, T., Müller, K.R., Samek, W. (eds.) xxAI 2020. LNCS, vol. 13200, pp. 13–38. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-04083-2_2
51. Humer, C., Hinterreiter, A., Leichtmann, B., Mara, M., Streit, M.: Comparing effects of attribution-based, example-based, and feature-based explanation methods on AI-assisted decision-making. Preprint, Open Science Framework (2022)

52. Ilyas, A., Park, S.M., Engstrom, L., Leclerc, G., Madry, A.: Datamodels: predicting predictions from training data. In: ICML (2022)
53. Jeyakumar, J.V., Noor, J., Cheng, Y.H., Garcia, L., Srivastava, M.: How can i explain this to you? An empirical study of deep neural network explanation methods. In: NeurIPS (2020)
54. Karimi, A.H., Barthe, G., Balle, B., Valera, I.: Model-agnostic counterfactual explanations for consequential decisions. In: AISTATS (2020)
55. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of StyleGAN. In: CVPR (2020)
56. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, Hoboken (2009)
57. Keane, M.T., Kenny, E.M.: The twin-system approach as one generic solution for XAI: an overview of ANN-CBR twins for explaining deep learning. In: IJCAI Workshop on XAI (2019)
58. Keane, M.T., Kenny, E.M., Delaney, E., Smyth, B.: If only we had better counterfactual explanations: five key deficits to rectify in the evaluation of counterfactual XAI techniques. arXiv preprint arXiv:2103.01035 (2021)
59. Keane, M.T., Smyth, B.: Good counterfactuals and where to find them: a case-based technique for generating counterfactuals for explainable AI (XAI). In: Watson, I., Weber, R. (eds.) ICCBR 2020. LNCS (LNAI), vol. 12311, pp. 163–178. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58342-2_11
60. Kenny, E.M., Keane, M.T.: Twin-systems to explain artificial neural networks using case-based reasoning: Comparative tests of feature-weighting methods in ANN-CBR twins for XAI. In: IJCAI (2019)
61. Kenny, E.M., Keane, M.T.: Explaining deep learning using examples: Optimal feature weighting methods for twin systems using post-hoc, explanation-by-example in XAI. KBS **233**, 107530 (2021)
62. Kenny, E.M., Keane, M.T.: On generating plausible counterfactual and semi-factual explanations for deep learning. In: AAAI (2021)
63. Khanna, R., Kim, B., Ghosh, J., Koyejo, S.: Interpreting black box predictions using fisher kernels. In: AISTATS (2019)
64. Kim, B., Khanna, R., Koyejo, O.O.: Examples are not enough, learn to criticize! Criticism for interpretability. In: NeurIPS (2016)
65. Kim, B., Rudin, C., Shah, J.A.: The Bayesian case model: a generative approach for case-based reasoning and prototype classification. In: NeurIPS (2014)
66. Kim, B., et al.: Interpretability beyond feature attribution: quantitative testing with concept activation vectors (TCAV). In: ICML (2018)
67. Kim, S., Jeong, M., Ko, B.C.: Lightweight surrogate random forest support for model simplification and feature relevance. Appl. Intell. **52**, 471–481 (2022)
68. Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. In: NeurIPS (2017)
69. Koh, P.W.W., Ang, K.S., Teo, H., Liang, P.S.: On the accuracy of influence functions for measuring group effects. In: NeurIPS (2019)
70. Kohonen, T.: The self-organizing map. IEEE (1990)
71. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**, 788–791 (1999)
72. Lewis, D.: Counterfactuals. Wiley, Hoboken (1973)
73. Lin, H., Bilmes, J.: Multi-document summarization via budgeted maximization of submodular functions. In: NAACL (2010)
74. Lin, H., Bilmes, J.: A class of submodular functions for document summarization. In: ACL HLT (2011)

75. Lipton, P.: Contrastive explanation. Roy. Inst. Philos. Supplements **27**, 247–266 (1990)
76. Marchette, C.E.P.D.J., Socolinsky, J.G.D.D.A.: Classification using class cover catch digraphs. J. Classif. **20**, 3-23 (2003)
77. Martens, J.: Deep learning via hessian-free optimization. In: ICML (2010)
78. Medin, D.L., Schaffer, M.M.: Context theory of classification learning. Psychol. Rev. **85**, 207 (1978)
79. Miller, T.: Explanation in artificial intelligence: insights from the social sciences. Artif. Intell. **267**, 1–38 (2019)
80. Miller, T.: Contrastive explanation: a structural-model approach. KER **36**, e14 (2021)
81. Ming, Y., Xu, P., Qu, H., Ren, L.: Interpretable and steerable sequence learning via prototypes. In: KDD (2019)
82. Mirzasoleiman, B., Karbasi, A., Badanidiyuru, A., Krause, A.: Distributed submodular cover: succinctly summarizing massive data. In: NeurIPS (2015)
83. Molnar, C.: Interpretable machine learning (2020). Lulu.com
84. Mothilal, R.K., Sharma, A., Tan, C.: Explaining machine learning classifiers through diverse counterfactual explanations. In: ACM FAccT (2020)
85. Nauta, M., Jutte, A., Provoost, J., Seifert, C.: This looks like that, because... explaining prototypes for interpretable image recognition. In: PKDD Workshop (2022)
86. Nauta, M., Van Bree, R., Seifert, C.: Neural prototype trees for interpretable fine-grained image recognition. In: CVPR (2021)
87. Ng, R.T., Han, J.: Efficient and effective clustering methods for spatial data mining. In: VLDB (1994)
88. Nosofsky, R.M.: Choice, similarity, and the context theory of classification. JEP LMC **10**, 104 (1984)
89. Nugent, C., Cunningham, P.: A case-based explanation system for black-box systems. Artif. Intell. Rev. **24**, 163–178 (2005)
90. Nugent, C., Doyle, D., Cunningham, P.: Gaining insight through case-based explanation. JIIS **32**, 267–295 (2009)
91. Olah, C., Mordvintsev, A., Schubert, L.: Feature visualization. Distill **2**, e7 (2017)
92. Papernot, N., McDaniel, P.: Deep k-nearest neighbors: towards confident, interpretable and robust deep learning. arXiv preprint arXiv:1803.04765 (2018)
93. Park, J.H., Im, K.H., Shin, C.K., Park, S.C.: MBNR: case-based reasoning with local feature weighting by neural network. Appl. Intell. **21**, 265–276 (2004)
94. Peng, Y., He, X., Zhao, J.: Object-part attention model for fine-grained image classification. IEEE TIP **27**, 1487–1500 (2017)
95. Pruthi, G., Liu, F., Kale, S., Sundararajan, M.: Estimating training data influence by tracing gradient descent. In: NeurIPS (2020)
96. Ribeiro, M.T., Singh, S., Guestrin, C.: Model-agnostic interpretability of machine learning. In: KDD (2016)
97. Roese, N.J., Olson, J.M.: Counterfactual thinking: a critical overview. What might have been: the social psychology of counterfactual thinking (1995)
98. Sani, S., Wiratunga, N., Massie, S.: Learning deep features for kNN-based human activity recognition. In: CEUR Workshop (2017)
99. Saunshi, N., Gupta, A., Braverman, M., Arora, S.: Understanding influence functions and datamodels via harmonic analysis. In: ICLR (2023)
100. Schank, R.C.: Dynamic Memory: A Theory of Reminding and Learning in Computers and People. Cambridge University Press, Cambridge (1983)

101. Schioppa, A., Zablotskaia, P., Vilar, D., Sokolov, A.: Scaling up influence functions. In: AAAI (2022)
102. Schoenborn, J.M., Weber, R.O., Aha, D.W., Cassens, J., Althoff, K.D.: Explainable case-based reasoning: a survey. In: AAAI Workshop (2021)
103. Schölkopf, B., Herbrich, R., Smola, A.J.: A generalized representer theorem. In: Helmbold, D., Williamson, B. (eds.) COLT 2001. LNCS (LNAI), vol. 2111, pp. 416–426. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44581-1_27
104. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: visual explanations from deep networks via gradient-based localization. In: ICCV (2017)
105. Serrurier, M., Mamalet, F., Fel, T., Béthune, L., Boissin, T.: When adversarial attacks become interpretable counterfactual explanations. arXiv preprint arXiv:2206.06854 (2022)
106. Shin, C.K., Yun, U.T., Kim, H.K., Park, S.C.: A hybrid approach of neural network and memory-based learning to data mining. IEEE TNN **11**, 637–646 (2000)
107. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: ICML (2017)
108. Simon, I., Snavely, N., Seitz, S.M.: Scene summarization for online image collections. In: ICCV (2007)
109. Sui, Y., Wu, G., Sanner, S.: Representer point selection via local Jacobian expansion for post-hoc classifier explanation of deep neural networks and ensemble models. In: NeurIPS (2021)
110. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: ICML (2017)
111. Szegedy, C., et al.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
112. Tversky, A.: Features of similarity. Psychol. Rev. **84**, 327 (1977)
113. Verma, S., Boonsanong, V., Hoang, M., Hines, K.E., Dickerson, J.P., Shah, C.: Counterfactual explanations and algorithmic recourses for machine learning: a review. arXiv preprint arXiv:2010.10596 (2020)
114. van der Waa, J., Nieuwburg, E., Cremers, A., Neerincx, M.: Evaluating XAI: a comparison of rule-based and example-based explanations. Artif. Intell. **291**, 103404 (2021)
115. Wachter, S., Mittelstadt, B., Russell, C.: Counterfactual explanations without opening the black box: automated decisions and the GDPR. Harv. JOLT **31**, 841 (2017)
116. Xu-Darme, R., Quénot, G., Chihani, Z., Rousset, M.C.: PARTICUL: part identification with confidence measure using unsupervised learning. arXiv preprint arXiv:2206.13304 (2022)
117. Yeh, C.K., Kim, J., Yen, I.E.H., Ravikumar, P.K.: Representer point selection for explaining deep neural networks. In: NeurIPS (2018)
118. Zhang, R., Madumal, P., Miller, T., Ehinger, K.A., Rubinstein, B.I.: Invertible concept-based explanations for CNN models with non-negative concept activation vectors. In: AAAI (2021)