Contextual Importance and Utility: a Theoretical Foundation *

Kary Främling

1 Department of Computing Science, Umeå University
Mit-huset, 901 87 Umeå, Sweden
kary.framling@umu.se
2 Department of Computer Science, Aalto University
Konemiehentie 1, 02150 Espoo, Finland

Abstract. This paper provides new theory to support the eXplainable AI (XAI) method Contextual Importance and Utility (CIU). CIU arithmetic is based on the concepts of Multi-Attribute Utility Theory, which gives CIU a solid theoretical foundation. The novel concept of contextual influence is also defined, which makes it possible to compare CIU directly with so-called additive feature attribution (AFA) methods for model-agnostic outcome explanation. One key takeaway is that the ‘influence’ concept used by AFA methods is inadequate for outcome explanation purposes even for simple models to explain. Experiments with simple models show that explanations using contextual importance (CI) and contextual utility (CU) produce explanations where influence-based methods fail. It is also shown that CI and CU guarantees explanation faithfulness towards the explained model.

Keywords: Explainable AI · Contextual Importance and Utility · Multi-Attribute Utility Theory · Decision Theory.

1 Introduction

Contextual Importance and Utility (CIU) was originally proposed by Kary Främling in 1995 in a context of Multiple Criteria Decision Making (MCDM). MCDM is a domain where mathematical models are used as Decision Support Systems (DSS) for human decision makers. No matter what model is being used for the DSS, it is crucial that the recommendations or outcome of the DSS can be presented in ways that are understandable for the decision makers, as well as for the people who might be affected by the decisions. CIU is model-agnostic and provides uniform explanation concepts for all possible DSS models, ranging from linear models such as the weighted sum, to rule-based systems, decision trees, fuzzy systems, neural networks and any machine learning-based models.

This paper solidifies and extends CIU theory and relates it to currently popular methods in the domain called eXplainable AI (XAI). In recent years, the

* The work is partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.
XAI domain has moved forward rapidly and has developed its own concepts and methods, which makes it difficult for current XAI researchers to understand and assess CIU in relation to their own work. The objectives of the paper are the following:

– Present a solid mathematical theory for CIU.
– Provide distinct definitions of the concepts influence, importance and utility.
– Define the new concept of contextual influence derived from CIU.
– Situate CIU within the latest state-of-the-art in XAI and show that it performs better than core mainstream XAI methods.

After this Introduction, Section 2 goes through the theoretical constructs of CIU and relates CIU to the family of additive feature attribution methods such as Shapley values and LIME. Section 3 provides empirical evidence for the theory in Section 2 followed by Conclusions.

2 Theory

2.1 Additive Feature Attribution Methods

We will here use notations from the paper by Lundberg and Lee [8] because it provides a unifying view on a whole family of outcome explanation methods called additive feature attribution (AFA) methods. Such methods use an explanation model $g$ that is an interpretable approximation of the original model $f$.

The following definition is fundamental for AFA methods [8]:

**Definition 1.** AFA methods have an explanation model that is a linear function of binary variables:

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z_i',$$

where $z' \in \{0, 1\}^M$, $M$ is the number of simplified input features, and $\phi \in \mathbb{R}$.

Methods with explanation models $g$ matching this definition attribute an effect $\phi_i$ to each input feature. Since $\phi_i$ is a scalar, the definition signifies that the explanation model $g$ is linear by definition. Our interpretation of the variable $\phi_i$ is to call it ‘influence’, which has also been used by other authors. Lundberg and Lee use the word ‘effect’ for $\phi_i$ but they do also use the word ‘importance’ with a similar meaning in [8]. It seems like most authors use ‘effect’, ‘influence’, ‘significance’, ‘importance’, etc. interchangeably.

The Shapley value is an AFA method originating from cooperative game theory [13]. The concept was picked up by the XAI community [14] and has become popular for producing outcome explanations following [8], and the introduction of SHAP (SHapley Additive exPlanations). The method distributes the difference between the prediction output $f(x)$ and the reference level $\phi_0$ to the input $x$.

\[^3\]Called baseline by many authors but ‘baseline’ seems to be used also for other purposes. This is why we prefer using ‘reference level’.
feature influences $\phi_i$ according to Equation 1. The most used $\phi_0$ value is the global average predicted value for the studied output in the training data set. If $f(x) > \phi_0$, then the sum of the terms $\phi_i$ must be positive, and vice versa if $f(x) < \phi_0$.

Local Interpretable Model-agnostic Explanations (LIME) is a popular AFA method, which creates a linear surrogate model $g$ that locally approximates the behaviour of the model to explain around the neighborhood of the instance being explained [12]. The sign of $\phi_i$ determines if the influence of the input feature $i$ is negative or positive. The magnitude of $\phi_i$ expresses how great the influence is.

2.2 Decision Theory and Multi-Attribute Utility Theory

In statistics, Decision Theory proposes a set of quantitative methods for reaching optimal, or at least rational, decisions. A decision problem must be capable of being formulated in terms of initial conditions and outcomes or courses of action, with their consequences. Each outcome is assigned a utility value based on the preferences of the decision maker(s). An optimal decision is one that maximizes the expected utility. It was proven already in 1947 that any individual whose preferences satisfy four axioms has a utility function, $u$, by which an individual's preferences can be represented on an interval scale [10].

If preferences over choices on attributes or input features $x_1, \ldots, x_n$ depend only on their marginal probability distributions, then the n-attribute utility function is additive according to:

$$u(x_1, \ldots, x_n) = \sum_{i=1}^{n} k_i u_i(x_i)$$ (2)

where $u$ and the $u_i$ are normalized to the range [0,1], and the $k_i$ are normalization constants [11]. If the goal is to simply rank-order the available choices, then a key condition for the additive form in Equation 2 is mutual preference independence. A fundamental result in utility theory is that two attributes are additive-independent if and only if their two-attribute utility function is additive and has the form:

$$u(x_1, x_2) = u(x_1) + u(x_2)$$ (3)

CIU respects additive-independence as long as the underlying model is linear. However, the objective of CIU is not to provide a rank-ordering but providing an explanation for the outcome of an underlying DSS model, which requires and justifies breaking the additive-independence condition given in Equation 3 as explained in the next section.

2.3 Contextual Importance and Utility (CIU)

CIU estimates the values $k_i$ and $u_i(y_i)$ in Equation 2 for one or more input features $\{i\}$ in a specific context $C$ and any black-box model $f$, where the context is defined by the instance or situation to be explained.
However, the use of Equation 2 makes it necessary to map output values \( y = f(x) \) into utility values \( u \) that are limited to the range \([0, 1]\). In classification tasks, the \( y \) values are usually probability values in the range \([0, 1]\) by definition, so it can be considered that \( u = y \). The same is not true for regression tasks. For instance, in the well-known Boston Housing data set, the output value is the median value of owner-occupied homes in $1000's and is in the range \([5, 50]\). A straightforward way of transforming that value into a utility value is an affine transformation \([5, 50] \mapsto [0, 1] \), assuming that the preference is to have a higher value. However, from a buyer’s point of view, the preference might be for lower prices and then the transformation would rather be \([50, 5] \mapsto [0, 1] \). In this paper, we will assume that \( u_j(y_j) \) is an affine transformation of the form \( u_j(y_j) = Ay_j + b \), where \( j \) is the output index. In practice, \( u_j(y_j) \) could have any shape as long as it produces values in the range \([0, 1]\) but that case goes beyond the scope of the current paper (and theory). This takes us to the definition of Contextual Importance (CI).

**Definition 2 (Contextual Importance).**

\[
CI_j(C, \{i\}, \{I\}) = \frac{umax_j(C, \{i\}) - umin_j(C, \{i\})}{umax_j(C, \{I\}) - umin_j(C, \{I\})},
\]

where \( \{i\} \subseteq \{I\} \) and \( \{I\} \subseteq \{1, \ldots, n\} \). \( C \) is the instance/context to be explained and defines the values of input features that do not belong to \( \{i\} \) or \( \{I\} \).

For clarity, \( \{i\} \) is the set of indices studied and \( \{I\} \) is the set of indices relative to which we calculate CI. When \( \{I\} = \{1, \ldots, n\} \), CI is calculated relative to the output utilities \( u_j \). For instance, \( CI_j(C, \{2\}, \{1, \ldots, n\}) \) is the contextual importance of input \( x_2 \), whereas \( CI_j(C, \{1, 2, 3\}, \{1, \ldots, n\}) \) is the joint contextual importance of inputs \( x_1, x_2, x_3 \) and \( CI_j(C, \{1, \ldots, n\}, \{1, \ldots, n\}) \) is the joint contextual importance of all inputs. \( umin_j() \) and \( umax_j() \) are the minimal and maximal utility values \( u_j \) observed for output \( j \) for all possible \( x_{\{i\}} \) and \( x_{\{I\}} \) values in the context \( C \), while keeping other input values at \( C \). Using \( \{I\} \neq \{1, \ldots, n\} \) makes it possible to also use and explain intermediate concepts as described in [234].

When \( u_j(y_j) = Ay_j + b \), then CI can be directly calculated as:

\[
CI_j(C, \{i\}, \{I\}) = \frac{ymax_j(C, \{i\}) - ymin_j(C, \{i\})}{ymax_j(C, \{I\}) - ymin_j(C, \{I\})},
\]

where \( ymin_j() \) and \( ymax_j() \) are the minimal and maximal \( y_j \) values observed for output \( j \). Equation 5 is identical to the CI definitions in [34].

The values of \( umin_j(\) and \( umax_j(\) can only be calculated exactly if the entire set of possible values for the input features \( \{i\} \) is available and the corresponding \( u_j \) values can be calculated in reasonable time. For categorical input features this is feasible as long as the number of possible values doesn’t grow too big. For continuous-valued input features, using a *Set of representative input vectors*
Algorithm 1: Set of representative input vectors

\[ S(C, \{i\}) \]

\[ \textbf{Result: } N \times M \text{ matrix } S(C, \{i\}) \]

1 begin
2 forall categorical input features do
3 \( D \leftarrow \) all possible value combinations for discrete inputs \( \{i\} \);
4 Randomize row order in \( D \);
5 if \( D \) has more rows than \( N \) then
6 Set \( N \) to number of rows in \( D \);
7 end
8 end
9 forall numerical input features do
10 Initialize \( N \times M \) matrix \( R \) with current input values \( C \);
11 \( R \leftarrow \) two rows per continuous-valued inputs in \( \{i\} \) where the current value is replaced by the values \( \min_{(i)} \) and \( \max_{(i)} \) respectively;
12 \( R \leftarrow \) fill remaining rows to \( N \) with random values from intervals \( [\min_{(i)}, \max_{(i)}] \);
13 end
14 \( S(C, \{i\}) \leftarrow \) concatenation of \( C \) with merged \( D \) and \( R \), where \( D \) is repeated if needed to obtain \( N \) rows;
15 end

\( S(C, \{i\}) \) is a model-agnostic approach to estimate \( \umin_j \) and \( \umax_j \). Algorithm 1 shows how \( S(C, \{i\}) \) is constructed in the ‘ciu’ R package \[4\]. The approach taken there is to limit the range of input values to intervals \( [\min_{(i)}, \max_{(i)}] \) for numerical input features. The studied instance \( C \) is the first sample in \( S(C, \{i\}) \), followed by samples with the extreme values \( \min_{(i)} \) and \( \max_{(i)} \) for numerical input features \( \{i\} \). For numerical input features, the remaining samples for achieving \( N \) samples are generated randomly from the interval(s) \( [\min_{(i)}, \max_{(i)}] \).

Other sampling methods could be envisaged, including model-specific ones like the one in Främling’s thesis \[3\] and remains a topic of future research.

When the set of input features to explain \( \{i\} \) is a subset of all input features \( \{1, \ldots, n\} \), then we apply the \textit{ceteris-paribus} principle, i.e. ‘other things held constant’ for estimating their CI value. This signifies that all input features \( \neg\{i\} \) are held constant at the values given by the studied instance \( C \) while estimating \( CI_j(C, \{i\}) \) by varying the values of the input features \( \{i\} \) according to Algorithm 1. This leads us to the following:

\textbf{Lemma 1 (Contextual Importance of input feature subsets \( \{i\} \)).}

When \( \{I\} \subseteq \{1, \ldots, n\} \) and \( \{i\} \subseteq \{I\} \) \( \Rightarrow \)
\[
[\umin_j(C, \{i\}), \umax_j(C, \{i\})] \subseteq [\umin_j(C, \{I\}), \umax_j(C, \{I\})].
\]

\textbf{Proof.} When \( \{i\} \subseteq \{I\} \), then \( S(C, \{i\}) \subseteq S(C, \{I\}) \) and
\[
[\umin_j(C, \{i\}), \umax_j(C, \{i\})] \subseteq [\umin_j(C, \{I\}), \umax_j(C, \{I\})]
\] when the number of samples \( N \to \infty \).

When considering that \( \umax_j(C, \{i\}) - \umin_j(C, \{i\}) \geq 0 \), we get:
Theorem 1 (Maximal range of Contextual Importance). $CI_j(C, \{i\}) \in [0, 1]$ for any set of input features $\{i\}$.

The Contextual Utility ($CU$) corresponds to the factor $u_j(x_i)$ in Equation 3. $CU$ expresses to what extent the current value of a given input feature contributes to obtaining a high output utility $u_j$.

Definition 3 (Contextual Utility).

$$CU_j(C, \{i\}) = \frac{u_j(C) - \min_j(C, \{i\})}{\max_j(C, \{i\}) - \min_j(C, \{i\})}$$ \hspace{1cm} (6)

When $u_j(y_j) = Ay_j + b$, this can be written as:

$$CU_j(C, \{i\}) = \frac{y_j(C) - \min_j(C, \{i\})}{\max_j(C, \{i\}) - \min_j(C, \{i\})},$$ \hspace{1cm} (7)

where $\min = \min$ if $A$ is positive and $\min = \max$ if $A$ is negative. This definition of $CU$ differs from CI definitions in $[3, 4]$ by handling negative $A$ values correctly.

Illustration of Additive Independence in CIU for linear model $f$. We will next illustrate that CIU satisfies Equations 2 and 3 when the input features $x_1, \ldots, x_n$ are additive-independent. For this, we use the simple function $y = x_1 + x_2$ with $x_i \in [0, 1]$. In this case we can use $u_i(x_i) = x_i$. Table 1 shows results for all the four zero-one combinations when $u(x_i) = CI(x_i) \times CU(x_i)$.

| $x_1$ | $x_2$ | $y = x_1 + x_2$ | $CI(x_1)$ | $CI(x_2)$ | $CU(x_1)$ | $CU(x_2)$ | $u(x_1) + u(x_2)$ | $u(y) = u(x_1, x_2)$ |
|-------|-------|----------------|-----------|-----------|-----------|-----------|-------------------|----------------------|
| 0     | 0     | 0              | 0.5       | 0.5       | 0         | 0         | 0                 | 0                    |
| 0     | 1     | 1              | 0.5       | 0.5       | 0         | 1         | 0.5               | 0.5                  |
| 1     | 0     | 1              | 0.5       | 0.5       | 1         | 0         | 0.5               | 0.5                  |
| 1     | 1     | 2              | 0.5       | 0.5       | 1         | 1         | 1                 | 1                    |

We now go to the core point of disruption of CIU with utility theory: most models $f$ for which we would like to provide explainability are non-linear and their input features tend to be dependent on each other. Therefore, CIU proposes to abandon the requirement of additive-independence of Equation 3. An initial assumption of CIU is indeed that both the importance and the utility function can (and usually do) depend on the values of other input features, which is the main reason for using the word contextual in CIU.

In order to illustrate the need and necessity to take the contextual aspects into account for outcome explanation, we will study how to explain results of
Table 2: OR function input and output values, with corresponding CI and CU values.

| x₁ | x₂ | y   | CI(x₁) | CU(x₁) | u(x₁) + u(x₂) | u(y) = u(x₁, x₂) |
|----|----|-----|--------|--------|---------------|------------------|
| 0  | 0  | 0   | 0      | 0      | 0             | 0                |
| 0  | 1  | 1   | 1      | 1      | NaN           | 1                |
| 1  | 0  | 0   | 1      | 0      | NaN           | 1                |
| 1  | 1  | 0   | 0      | NaN    | NaN           | 0                |

Table 3: XOR function input and output values, with corresponding CI and CU values.

| x₁ | x₂ | y   | CI(x₁) | CU(x₁) | u(x₁) + u(x₂) | u(y) = u(x₁, x₂) |
|----|----|-----|--------|--------|---------------|------------------|
| 0  | 0  | 0   | 0      | 0      | 0             | 0                |
| 0  | 1  | 1   | 1      | 0      | 0             | 2                |
| 1  | 0  | 0   | 1      | 1      | 0             | 2                |
| 1  | 1  | 1   | 0      | 0      | 0             | 0                |

simple OR and XOR functions, where the input features are clearly dependent. The results are shown in Tables 2 and 3.

A core reason for showing these three simple examples is to emphasize that both CI and CU are absolute values in the range [0, 1], as opposed to relative values used by AFA methods. CI_j(C, {i}) = 0 signifies that in the context C the input feature(s) {i} have no effect on the utility u_j of output j. CI_j(C, {i}) = 1 signifies that changes to the values of input feature(s) {i} can modify the value of u_j over the entire range [0, 1]. Similarly, CU_j(C, {i}) = 0 signifies that the value(s) of input feature(s) {i} are the least favorable (in the sense of utility u_j) for the output j. CU_j(C, {i}) = 1 signifies that the value(s) of input feature(s) {i} are the most favorable for the output j.

2.4 Contextual influence

CI and CU produce explanations from any model f in a uniform way, no matter if the model is linear or not, continuous-valued or discrete, hand-coded or created via machine learning. However, in order to compare CIU with AFA methods, we define Contextual influence. We begin by a contextual version of the term k_iu_i(x_i) in Equation 2

\[ Cinfluence_j(C, \{i\}) = CI_j(C, \{i\}) \times CU_j(C, \{i\}) \]

when k_i = CI_j(C, {i}) and u_i = CU_j(C, {i}). Cinfluence can be scaled into any range [rmin, rmax], which leads us to the following definition:

Definition 4 (Contextual influence).

\[ \phi = (rmax - rmin) \times CI \times (CU - neutral.CU) \] (8)
where \( j(C, \{i\}) \) has been omitted from all three terms \( \phi, CI, \) and \( CU \) for easier readability.

The symbol \( \phi \) has been chosen on purpose to signify ‘influence’ as for Shapley values and LIME. We use \([r_{\min}, r_{\max}] = [-1, 1]\) in Section 3. Setting \( \text{neutral.CU} = 0.5 \) restricts \( \phi \) values to only negative, zero or positive, as for Shapley values and LIME.

2.5 CIU versus Additive Feature Attribution Methods

As shown in the previous Sections, CIU makes a clear distinction between ‘importance’ and ‘influence’. Furthermore, CIU uses the notions of ‘utility function’ and ‘utility’, which are not considered by any known AFA method. As shown by the following differences, CIU is not an AFA method:

- CIU does not use or create any explanation model \( g \).
- CI and CU can be used for calculating an influence measure \( \phi \) but it is not possible to do it the other way around.
- CI and CU provide absolute values in the range \([0, 1]\) that have precise definitions, whereas \( \phi \) values express relative influence between input features.
- CIU is defined using utility theory and CIU explanations are entirely based on elements of that theory. CIU does not attempt to mimic or approximate the original function \( f \) in any way.
- CIU has no notion \( z_i' \) of presence or not of an input feature and should not be confused with so-called occlusion-based methods [15].

Intuitively, it might be possible to consider Contextual influence in Equation 3 to be an AFA method, which is one reason for using the symbol \( \phi \) for it. However, the fact that the reference level \( \text{neutral.CU} \) is defined on utility values \( u_j \) and not on output values \( y_j \) as in AFA methods is a major difference. Furthermore, there’s no additivity requirement on Contextual influence, even though additivity could be imposed by normalization. Still, further research on comparing Contextual influence and AFA methods is interesting and ongoing.

3 Experimental Evaluation

In this section we compare CIU, contextual influence, Shapley values and LIME for three known functions that have two input features \( x_1, x_2 \) and one output value \( y \). The functions are linear (\( y = 0.3x_1 + 0.7x_2 \)), rule-based and ‘sombrero’ (\( y = \sin(\sqrt{x_1^2 + x_2^2})/\sqrt{x_1^2 + x_2^2} \)), as shown in Figure 1. The studied input values \( C = (x_1, x_2) \) are indicated by the red dots in Figure 1. Figure 2 shows how the output \( y \) changes as a function of one input feature while keeping constant the value of the other input feature, together with values and illustrations of CIU calculations.

CIU results are produced using the ‘ciu’ R package [16]. Shapley values are produced with the ‘IML’ R package [9] and LIME results are produced with the
‘lime’ R package [11]. All methods were run with default parameters ($N = 100$ for CIU). In Figure 3 the order of input features is determined automatically by the respective package, so it is not necessarily the same in all bar plots.

Fig. 1: Linear, rule-based and non-linear models used in the study.

CIU can use the test functions directly as the model to explain and does not need a data set. Both Shapley values and LIME do require a training data set, which has been generated as a regular grid of points $(x_1, x_2)$. A grid $x_1, x_2 \in [0, 1]$ with step size 0.05 is used for the linear and rule-based functions. For the ‘sombrero’ function, $x_1, x_2 \in [-10, 10]$ with step size 0.51. Table 4 and Figure 3 show the results of the different methods. The following observations are made:

CIU consistently describes a) how much the utility (‘goodness’) of $y$ can change when modifying the value of $x_i$ from the least favorable to the most favorable, and b) how favorable the current value $x_i$ is for the utility $u(y)$, for the current instance/context $C$. CIU shows complete fidelity towards the underlying model, as indicated in Table 4 which correspond exactly to the weights and input/utility values for the linear function. CI and CU values can also be ‘seen’ and understood directly from the graphs in Figure 2.

Influence-based explanations are inconsistent between the different methods for the three last test cases. In particular for the linear function with $(x_1, x_2) = (0.5, 0.5)$, the influence-based explanations are in-existent because all $\phi_i$ values are (or should be) zero, which is indeed the case for all three influence-based methods. The slight deviations from zero are only due to numerical imprecision.

Table 4: Results for known functions with two inputs and one output.

| $f(x)$ | $x_1$ | $x_2$ | $y$ | $CI_1$ | $CI_2$ | $CU_1$ | $CU_2$ | $\phi_{1}^{lin}$ | $\phi_{2}^{lin}$ | $\phi_{1}^{rule}$ | $\phi_{2}^{rule}$ | $\phi_{1}^{sombrero}$ | $\phi_{2}^{sombrero}$ |
|--------|-------|-------|-----|--------|--------|--------|--------|-----------------|-----------------|-----------------|-----------------|-------------------|-------------------|
| Linear | 0.7   | 0.8   | 0.77| 0.7    | 0.7    | 0.8    | 0.12   | 0.42 0.065      | 0.208           | 0.040 0.331     |                  |                   |
| Linear | 0.5   | 0.5   | 0.5  | 0.5    | 0.5    | 0.5    | 0.0    | 0.0   0.007 -0.021 | -0.054         | -0.097         |                  |                   |
| Rules  | 0.7   | 0.4   | 0.6  | 0.8    | 1.0    | 0.5    | 0.6    | 0.0   0.218 -0.046 | 0.285          | -0.117         |                  |                   |
| Sombrero| -7.5  | -1.5  | 0.128| 0.724  | 0.18   | 0.392  | 0.998  | -0.157 0.18 0.061 | 0.032         | -0.019 0.010    |                  |
for contextual influence, whereas sampling leads to stochastic \( \phi \) values that are normally distributed around zero for Shapley values and LIME.

Contextual influence shows greater stability than the other influence-based methods. It also remains possible to ‘read’ and understand the contextual influence values directly from Figure 2 because it corresponds to \( y - \text{neutral.CU} \) relative to the \([y_{\text{min}}, y_{\text{max}}] \) range.

Shapley values distribute the difference \( f(x) - \phi_0 \) ‘fairly’ over all inputs \( \phi_i \). However, when that difference is zero, as for the linear function with \((x_1, x_2) = (0.5, 0.5)\), then there is nothing to distribute, at least not to input features with identical and ‘average’ values, so Shapley values fail in producing any explanation in this case. The corresponding bar plot in Figure 3 gives an impression that there is a successful explanation but that is because influence values are relative, so the scale of the \( x \)-axis is extended. In practice, both \( \phi \) values are (or should be) zero, as seen in Table 1.

LIME. For some reason, LIME’s ‘Explanation fit’ is 0.5 for the linear function with \((x_1, x_2) = (0.7, 0.8)\) but only 0.06 with \((x_1, x_2) = (0.5, 0.5)\). It also remains unclear what is actually the reference level \( \phi_0 \) used by LIME.

The experiments shown in this paper emphasize the theoretical difference between CIU and AFA methods, as well as the difference between the concepts of importance, utility and influence. All methods are applicable to any real-world tabular data sets and an extensive comparison between CIU, LIME and Shapley values is presented in [6]. The CIU Github site [https://github.com/KaryFramling/ciu](https://github.com/KaryFramling/ciu) provides executable examples at least for the well-known

Fig. 2: Output value as a function of one variable for the test functions, with illustration of how CI and CU are calculated.
benchmark data sets Iris, Boston, Heart Disease, UCI Cars, Diamonds, Titanic and Adult and several different machine learning models. CIU is also implemented for image explanations as reported in [5,7]. The source code used in this paper is published at [https://github.com/KaryFramling/AJCAI_2021](https://github.com/KaryFramling/AJCAI_2021).

4 Conclusion

This paper extends the theory of CIU beyond the original theory in [3] and defines the new concept of contextual influence. CIU is compared to current state-of-the-art methods, notably the family of AFA methods. As shown in the paper, CIU provides new flexibility and expressiveness by separating the notions of importance and utility from the notion of influence used by AFA methods. It is also illustrated why ‘influence’ alone lacks in explanation capability. Identified advantages of CIU compared to AFA methods are:

1. CI and CU provide absolute values in the [0, 1] range that have clear definitions and interpretations.
2. Separate ‘importance’ and ‘utility’ concepts allow for more fine-grained and accurate explanations than ‘influence’ alone.
3. CIU has only one ‘tunable’ parameter (the number $N$ of samples to use), which provides robustness and simplicity of use.
4. CIU does not need access to a data set.
5. CIU is not a ‘black box’ in itself because CI and CU values can be ‘read out’ directly from input versus output graphs.
6. CIU’s faithfulness/fidelity towards the model $f$ is guaranteed because no interpretable model $g$ is needed. CIU’s faithfulness only depends on how accurately $umin$ and $umax$ can be estimated.
7. CIU has a solid and proven mathematical background and framework in multi-attribute utility theory, which puts it at least on the same level of rigor as Shapley values.
References

1. Dyer, J.S.: Maut — Multiattribute Utility Theory, pp. 265–292. Springer New York, New York, NY (2005)
2. Främling, K.: Explaining results of neural networks by contextual importance and utility. In: Andrews, R., Diederich, J. (eds.) Rules and networks: Proceedings of the Rule Extraction from Trained Artificial Neural Networks Workshop, AISB’96 conference. Brighton, UK (1-2 April 1996)
3. Främling, K.: Modélisation et apprentissage des préférences par réseaux de neurones pour l’aide à la décision multicritère. Phd thesis, INSA de Lyon (Mar 1996)
4. Främling, K.: Contextual importance and utility in R: the 'ciu' package. In: Proceedings of 1st Workshop on Explainable Agency in Artificial Intelligence, at 35th AAAI Conference on Artificial Intelligence, February 2-9, 2021. pp. 110–114 (2021)
5. Främling, K., Knapič, S., Malhi, A.: ciu.image: An R package for explaining image classification with contextual importance and utility. In: Calvaresi, D., Najjar, A., Winikoff, M., Främling, K. (eds.) Explainable and Transparent AI and Multi-Agent Systems - 3rd International Workshop, EXTRAAMAS 2021. pp. 55–62. Lecture Notes in Computer Science, Springer, Germany (2021)
6. Främling, K., Westberg, M., Jullum, M., Madhimeri, M., Malhi, A.: Comparison of contextual importance and utility with lime and shapley values. In: Calvaresi, D., Najjar, A., Winikoff, M., Främling, K. (eds.) Explainable and Transparent AI and Multi-Agent Systems - 3rd International Workshop, EXTRAAMAS 2021. pp. 39–54. Lecture Notes in Computer Science, Springer, Germany (2021)
7. Knapič, S., Malhi, A., Saluja, R., Främling, K.: Explainable artificial intelligence for human decision support system in the medical domain. Machine Learning and Knowledge Extraction 3(3), 740–770 (2021)
8. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 4765–4774. Curran Associates, Inc. (2017)
9. Molnar, C., Casalicchio, G., Bischl, B.: iml: An R package for interpretable machine learning. J. Open Source Softw. 3(26), 786 (2018)
10. von Neumann, J., Morgenstern, O.: Theory of games and economic behavior. Princeton University Press (1947)
11. Pedersen, T.L., Benesty, M.: lime: Local Interpretable Model-Agnostic Explanations (2019), https://CRAN.R-project.org/package=lime r package version 0.5.1
12. Ribeiro, M.T., Singh, S., Guestrin, C.: ”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. pp. 1135–1144 (2016)
13. Shapley, L.: A value for n-person games. In: Kuhn, H., Tucker, A. (eds.) Contributions to the Theory of Games, Vol. II, Annals of Mathematics Studies, vol. 28, pp. 307–317. Princeton University Press, Princeton, NJ (1953)
14. Strumbelj, E., Kononenko, I.: An efficient explanation of individual classifications using game theory. J. Mach. Learn. Res. 11, 1–18 (Mar 2010)
15. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 818–833. Springer International Publishing, Cham (2014)