A Unifying Formal Approach to Importance Values in Boolean Functions

Hans Harder[1,2] , **Simon Jantsch**[2] , **Christel Baier**[2,4] and **Clemens Dubslaff**[3,4]
[1]University of Paderborn
[2]Dresden University of Technology
[3]Eindhoven University of Technology
[4]Centre for Tactile Internet with Human-in-the-Loop (CeTI)
hans.harder@uni-paderborn.de, {simon.jantsch, christel.baier}@tu-dresden.de, c.dubslaff@tue.nl

Abstract

Boolean functions and their representation through logics, circuits, machine learning classifiers, or binary decision diagrams (BDDs) play a central role in the design and analysis of computing systems. Quantifying the relative impact of variables on the truth value by means of *importance values* can provide useful insights to steer system design and debugging. In this paper, we introduce a uniform framework for reasoning about such values, relying on a generic notion of *importance value functions (IVFs)*. The class of IVFs is defined by axioms motivated from several notions of importance values introduced in the literature, including Ben-Or and Linial's *influence* and Chockler, Halpern, and Kupferman's notion of *responsibility* and *blame*. We establish a connection between IVFs and game-theoretic concepts such as *Shapley* and *Banzhaf* values, both of which measure the impact of players on outcomes in cooperative games. Exploiting BDD-based symbolic methods and projected model counting, we devise and evaluate practical computation schemes for IVFs.

1 Introduction

Boolean functions arise in many areas of computer science and mathematics, e.g., in circuit design, formal logics, coding theory, artificial intelligence, machine learning, and system analysis [Crama and Hammer, 2011a; O'Donnell, 2014]. When modeling and analyzing systems through Boolean functions, many design decisions are affected by the relevance of variables for the outcome of the function. Examples include noise-reduction components for important input variables to increase reliability of circuits, prioritizing important variables in decision-making of protocols, or the order of variables in BDDs [Bryant, 1992; Bartlett and Andrews, 2001]. Many ideas to quantify such notions of *importance* of variables in Boolean functions have since been considered in the literature. To mention a few, *influence* [Ben-Or and Linial, 1985] is used to determine power of actors in voting schemes, [Hammer *et al.*, 2000] devised measures based on how constant a function becomes depending on variable assignments, *blame* [Chockler and Halpern, 2004] quantifies the average *responsibility* [Chockler *et al.*, 2008] of input variables on the outcome of circuits or on causal reasoning, and the *Jeroslow-Wang value* [Jeroslow and Wang, 1990] quantifies importance of variables in CNFs to derive splitting rules for SAT-solvers [Hooker and Vinay, 1995]. Closely related are notions of impact in cooperative games, e.g., through the *Shapley value* [Shapley, 1953] or the *Banzhaf value* [Banzhaf, 1965].

Although some of the aforementioned concepts are of quite different nature and serve different purposes, they share some common ideas. This raises the question of what characteristics importance values have and how the notions of the literature relate. The motivation of this paper is to advance the understanding of importance values, independent of concrete applications. For this purpose, we introduce a generic axiomatic framework that constitutes the class of *importance value functions (IVFs)*. Our axioms are motivated by properties one would intuitively expect from IVFs, e.g., that independent variables have no importance or that permutations do not change importance values. We show basic relationships within and between IVFs and provide new insights for existing and new importance measures. By connecting Boolean functions and cooperative games through *cooperative game mappings (CGMs)* and using Shapley and Banzhaf values, we show how to generically derive new IVFs . All aforementioned notions of importance values from the literature satisfy our IVF axioms, showing that we provide a *unifying framework* for all these notions, including CGM-derived ones.

Most notions of importance are known to be computationally hard, e.g., computing influence or the Shapley value is #P-complete [Traxler, 2009; Faigle and Kern, 1992; Deng and Papadimitriou, 1994]. We address computational aspects by devising practical computation schemes for IVFs using projected model counting [Aziz *et al.*, 2015] and BDDs.

Contributions and outline. In summary, our main contribution is an axiomatic definition of IVFs for variables in Boolean functions (Section 3), covering notions of importance from the literature (Sections 4.1 and 4.2). Moreover, we derive novel IVFs by linking Boolean functions with cooperative games and related values (Section 4.3). Finally, we provide practical computation schemes for IVFs (Section 5).

Supplemental material. All proofs can be found in an extended version at https://arxiv.org/abs/2305.08103. An implementation of the computing schemes for IVFs can be found at https://github.com/graps1/impmeas.

2 Preliminaries

Let $X = \{x, y, z, \dots\}$ be a finite set of $n = |X|$ variables, which we assume to be fixed throughout the paper.

Assignments. An *assignment over* $U \subseteq X$ is a function $\boldsymbol{u}\colon U \to \{0,1\}$, written in the form $\boldsymbol{u} = x/0; y/1; \dots$. We denote assignments by bold lower-case letters and their domains by corresponding upper-case letters. If $\boldsymbol{u}$ and $\boldsymbol{v}$ have disjoint domains, we write their *concatenation* as $\boldsymbol{w} = \boldsymbol{u};\boldsymbol{v}$ with $W = V \cup U$ and $\boldsymbol{w}(x) = \boldsymbol{u}(x)$ if $x \in U$ and $\boldsymbol{w}(x) = \boldsymbol{v}(x)$ if $x \in V$. The *restriction* of $\boldsymbol{u}$ to a domain $S \subseteq U$ is denoted by $\boldsymbol{u}_S$. For a permutation σ of X, we define $\sigma\boldsymbol{u}$ as the assignment over $\sigma(U)$ with $(\sigma\boldsymbol{u})(x) = \boldsymbol{u}(\sigma^{-1}(x))$.

Boolean functions. We call $f, g, h, \dots : \{0,1\}^X \to \{0,1\}$ *Boolean functions*, collected in a set $\mathbb{B}(X)$. We write $g = x$ if g is the indicator function of x, and we write $\overline{g}$ for negation, $f \vee g$ for disjunction, fg for conjunction and $f \oplus g$ for exclusive disjunction. The *cofactor of f w.r.t. an assignment* $\boldsymbol{v}$ is the function $f_{\boldsymbol{v}}$ that always sets variables in V to the value given by $\boldsymbol{v}$, and is defined as $f_{\boldsymbol{v}}(\boldsymbol{u}) = f(\boldsymbol{v}; \boldsymbol{u}_{U \setminus V})$. The *Shannon decomposition of f w.r.t. variable x* is a decomposition rule stating that $f = x f_{x/1} \vee \overline{x} f_{x/0}$ holds, where $f_{x/1}$ and $f_{x/0}$ are the *positive* and *negative* cofactor of f w.r.t. x. For a Boolean function f, variable x, and Boolean function or variable s, let $f[x/s] = s f_{x/1} \vee \overline{s} f_{x/0}$ be the function that replaces x by s. For example, if $f = y \vee xz$, then $f_{x/1} = y \vee z$ and $f_{x/0} = y$. Moreover, for $s = x_1x_2$, we have

$$f[x/s] = s(y \vee z) \vee \overline{s}y = y \vee sz = y \vee x_1x_2z.$$

For $\sim \in \{\leq, \geq, =\}$, we write $f \sim g$ if $f(\boldsymbol{u}) \sim g(\boldsymbol{u})$ is true for all assignments. We collect the variables that f depends on in the set $\mathtt{dep}(f) = \{x \in X : f_{x/1} \neq f_{x/0}\}$. If $\boldsymbol{v}$ is an assignment with $\mathtt{dep}(f) \subseteq V$, then $f(\boldsymbol{v})$ denotes the only possible value that $f_{\boldsymbol{v}}$ can take.

We say that f is *monotone in* x if $f_{x/1} \geq f_{x/0}$, and call f *monotone* if f is monotone in all of its variables. Furthermore, f is the *dual* of g if $f(\boldsymbol{u}) = \overline{g}(\overline{\boldsymbol{u}})$, where $\overline{\boldsymbol{u}}$ is the variable-wise negation of $\boldsymbol{u}$. We call f *symmetric* if $f = \sigma f$ for all permutations σ of X, where $\sigma f(\boldsymbol{u}) = f(\sigma^{-1}\boldsymbol{u})$.

Expectations. We denote the expectation of f w.r.t. the uniform distribution over D by $\mathbb{E}_{d \in D}[f(d)]$ for $f : D \to \mathbb{R}$. We only consider cases where D is finite, so

$$\mathbb{E}_{d \in D}[f(d)] = \frac{1}{|D|} \sum_{d \in D} f(d).$$

If the domain of f is clear, we simply write $\mathbb{E}[f]$. For $f \in \mathbb{B}(X)$, $\mathbb{E}[f]$ is the fraction of satisfying assignments of f.

Modular decompositions. We introduce a notion of *modularity* to capture independence of subfunctions as common in the theory of Boolean functions and related fields [Ashenhurst, 1957; Birnbaum and Esary, 1965; Shapley, 1967; Bioch, 2010]. Intuitively, f is modular in g if f treats g like a subfunction and otherwise ignores all variables that g depends on. We define modularity in terms of a *template function* ℓ in which g is represented by a variable x:

Definition 1. Let $f, g \in \mathbb{B}(X)$. We call f *modular in* g if g is not constant and there is $\ell \in \mathbb{B}(X)$ and $x \in X$ such that $\mathtt{dep}(\ell) \cap \mathtt{dep}(g) = \varnothing$ and $f = \ell[x/g]$. If ℓ is monotone in x, then f is *monotonically modular in* g.

If f is modular in g with ℓ and x as above, then $f(\boldsymbol{u}) = \ell(\boldsymbol{w})$, where $\boldsymbol{w}$ is defined for $y \in X$ as

$$\boldsymbol{w}(y) = \begin{cases} g(\boldsymbol{u}) & \text{if } y = x, \text{ and} \\ \boldsymbol{u}(y) & \text{otherwise.} \end{cases}$$

Thus, the value computed by g is assigned to x and then used by ℓ, which otherwise is not influenced by the variables that g depends on. For example, $f = x_1 \vee z_1z_2x_2$ is modular in $g = z_1z_2$ since f can be obtained by replacing x in $\ell = x_1 \vee xx_2$ by g. Note that $\mathtt{dep}(\ell) = \{x, x_1, x_2\}$ and $\mathtt{dep}(g) = \{z_1, z_2\}$ are disjoint. This property is crucial, since it ensures f and g are coupled through variable x only.

If f is modular in g, then the cofactors $\ell_{x/1}$ and $\ell_{x/0}$ must be unique since g is not constant. Hence, we can define the *cofactors of f w.r.t. g* as $f_{g/1} = \ell_{x/1}$ and $f_{g/0} = \ell_{x/0}$. The instantiation is reversed by setting $f[g/x] = x f_{g/1} \vee \overline{x} f_{g/0}$.

Boolean derivatives. We frequently rely on the *derivative of a Boolean function f w.r.t. variable x,*

$$\mathrm{D}_x f = f_{x/1} \oplus f_{x/0},$$

which encodes the undirected change of f w.r.t. x. For example, $f = x \vee y$ has the derivative $\mathrm{D}_x f = \overline{y}$, with the intuition that x can only have an impact if y is set to zero. Furthermore, if f is modular in g, we define the *derivative of f w.r.t. g* as $\mathrm{D}_g f = f_{g/1} \oplus f_{g/0}$. Given this, we obtain the following lemma corresponding to the chain rule known in calculus:

Lemma 1. *Let f be modular in g and $x \in \mathtt{dep}(g)$. Then*

$$\mathrm{D}_x f = (\mathrm{D}_x g)(\mathrm{D}_g f).$$

3 Importance Value Functions

In this section, we devise axiomatic properties that should be fulfilled by every *reasonable* importance attribution scheme.

For a Boolean function f and a variable x, we quantify the importance of x in f by a number $\mathfrak{I}_x(f) \in \mathbb{R}$, computed by some *value function* $\mathfrak{I}$. Not every value makes intutive sense when interpreted as the "importance" of x, so we need to pose certain restrictions on $\mathfrak{I}$.

We argue that $\mathfrak{I}$ should be bounded, with 1 marking the highest and 0 the lowest importance; that functions which are independent of a variable should rate these variables the lowest importance (e.g., $\mathfrak{I}_x(f) = 0$ if $f = y \vee z$); that functions which depend on one variable only should rate these variables the highest importance (e.g., $\mathfrak{I}_x(f) = 1$ for $f = x$); that neither variable names nor polarities should play a role in determining their importance (e.g., $\mathfrak{I}_x(x\overline{z}) = \mathfrak{I}_z(x\overline{z})$, cf. [Slepian, 1953; Golomb, 1959]):

Definition 2 (IVF)**.** A *value function* is a mapping of the form $\mathfrak{I}\colon X \times \mathbb{B}(X) \to \mathbb{R}$ with $(x, f) \mapsto \mathfrak{I}_x(f)$. An *importance value function (IVF)* is a value function $\mathfrak{I}$ where for all $x, y \in X$, permutations $\sigma\colon X \to X$, and $f, g, h \in \mathbb{B}(X)$:

(BOUND) $0 \leq \mathfrak{I}_x(f) \leq 1$.

(DUM) $\mathfrak{I}_x(f) = 0$ if $x \notin \mathtt{dep}(f)$.

(DIC) $\mathfrak{I}_x(x) = \mathfrak{I}_x(\overline{x}) = 1$.

(TYPE) (i) $\mathfrak{I}_x(f) = \mathfrak{I}_{\sigma(x)}(\sigma f)$ and
(ii) $\mathfrak{I}_x(f) = \mathfrak{I}_x(f[y/\overline{y}])$.

(MODEC) $\mathfrak{I}_x(f) \geq \mathfrak{I}_x(h)$ if
(i) f and h are monotonically modular in g,
(ii) $f_{g/1} \geq h_{g/1}$ and $h_{g/0} \geq f_{g/0}$, and
(iii) $x \in \mathtt{dep}(g)$.

BOUND, DUM for "dummy", DIC for "dictator" and TYPE for "type invariance" were discussed above. MODEC (for "modular encapsulation consistency") is the only property that allows the inference of non-trivial importance inequalities in different functions. Let us explain its intuition. We say that f *encapsulates* h *on* g if these functions satisfy (i) and (ii) from MODEC. Intuitively, together with (i), condition (ii) states that *if one can control the output of g, it is both easier to satisfy f than h* (using $f_{g/1} \geq h_{g/1}$) *and to falsify f than* h (using $h_{g/0} \geq f_{g/0}$). We argue in MODEC that if f encapsulates h on g, then g's impact on f is higher than on h, and thus, the importance of variables in $\mathtt{dep}(g)$ (cf. (iii)) should be also higher w.r.t. f than w.r.t. h.

Example. Let $f = x_1x_2 \vee x_3x_4x_5$, $h = x_3x_4 \vee x_1x_2x_5$, and $\mathfrak{I}$ be an IVF. Then f encapsulates h on $g = x_1x_2$, since

$$\underbrace{1}_{f_{g/1}} \geq \underbrace{x_3x_4 \vee x_5}_{h_{g/1}} \geq \underbrace{x_3x_4}_{h_{g/0}} \geq \underbrace{x_3x_4x_5}_{f_{g/0}}.$$

We then get $\mathfrak{I}_{x_1}(f) \geq \mathfrak{I}_{x_1}(h)$ by application of MODEC. Swapping x_1 with x_3 and x_2 with x_4, we obtain a permutation σ such that $h = \sigma f$. By TYPE, we derive $\mathfrak{I}_{x_1}(h) = \mathfrak{I}_{x_3}(f)$. Using TYPE on the other variables yields

$$\mathfrak{I}_{x_1}(f) = \mathfrak{I}_{x_2}(f) \geq \mathfrak{I}_{x_3}(f) = \mathfrak{I}_{x_4}(f) = \mathfrak{I}_{x_5}(f).$$

Together with TYPE, MODEC implies the *Winder preorder*, which is similar in spirit (see [Hammer *et al.*, 2000]). However, MODEC generalizes to modular decompositions and allows inferring importance inequalities w.r.t. to different functions.

Biased and unbiased. We say that an IVF is *unbiased* if $\mathfrak{I}_x(g) = \mathfrak{I}_x(\overline{g})$ holds for all Boolean functions g and variables x. That is, unbiased IVFs measure the impact of variables without any preference for one particular function outcome, while biased ones quantify the impact to enforce a function to return one or zero. Biased IVFs can, e.g., be useful when the task is to assign responsibility values for the violation of a specification.

3.1 Further Properties

We defined IVFs following a conservative approach, collecting minimal requirements on IVFs. Further additional properties can improve on the predictability and robustness of IVFs.

Definition 3. A value function $\mathfrak{I}$ is called

- *rank preserving*, if for all $f, g \in \mathbb{B}(X)$ such that f is modular in g and $x, y \in \mathtt{dep}(g)$:
$$\mathfrak{I}_x(g) \geq \mathfrak{I}_y(g) \implies \mathfrak{I}_x(f) \geq \mathfrak{I}_y(f),$$
- *chain-rule decomposable*, if for all $f, g \in \mathbb{B}(X)$ such that f is modular in g and $x \in \mathtt{dep}(g)$:
$$\mathfrak{I}_x(f) = \mathfrak{I}_x(g)\mathfrak{I}_g(f),$$
where $\mathfrak{I}_g(f) = \mathfrak{I}_{x_g}(f[g/x_g])$ for some $x_g \notin \mathtt{dep}(f)$,
- and *derivative dependent*, if for all $f, g \in \mathbb{B}(X)$, $x \in X$:
$$\mathrm{D}_x f \geq \mathrm{D}_x g \implies \mathfrak{I}_x(f) \geq \mathfrak{I}_x(g).$$

We also consider *weak* variants of *rank preserving* and *chain-rule decomposable* where f ranges only over functions that are *monotonically* modular in g.

Rank preservation. Rank preservation states that the relation between two variables should not change if the function is embedded somewhere else. This can be desired, e.g., during a modeling process in which distinct Boolean functions are composed or fresh variables added, where rank preserving IVFs maintain the relative importance order of variables. We see this as a useful but optional property of IVFs since an embedding could change some parameters of a function that might be relevant for the relationship of both variables. For example, if $f = gz$ with $z \notin \mathtt{dep}(g)$, then the relative number of satisfying assignments is halved compared to g. If x is more important than y in g but highly relies on g taking value one, it might be that this relationship is reversed for f (cf. example given in Section 4.1).

Chain-rule decomposability. If an IVF is chain-rule decomposable, then the importance of a variable in a module is the product of (i) its importance w.r.t. the module and (ii) the importance of the module w.r.t. the function. Many values studied in this paper satisfy this property (Section 4).

Example. Let $f = x_1 \oplus \cdots \oplus x_m$, and let $\mathfrak{I}$ be a chain-rule decomposable IVFs with $\mathfrak{I}_x(x \oplus y) = \alpha$. Since f is modular in $g = x_1 \oplus \cdots \oplus x_{m-1}$, and g modular in $x_1 \oplus \cdots \oplus x_{m-2}$, etc., we can apply the chain-rule property iteratively to get

$$\mathfrak{I}_{x_1}(f) = \mathfrak{I}_{x_1}(g)\mathfrak{I}_g(f) = \mathfrak{I}_{x_1}(g)\alpha = \cdots = \alpha^{m-1},$$

where we use TYPE to derive $\mathfrak{I}_g(f) = \mathfrak{I}_{x_g}(x_g \oplus x_m) = \alpha$.

Derivative dependence. Derivative dependence states that an IVF should quantify the *change* a variable induces on a Boolean function. It can be used to derive, e.g., the inequality $\mathfrak{I}_{x_1}(x_1 \oplus x_2x_3) \geq \mathfrak{I}_{x_1}(x_2 \oplus x_1x_3)$, which is not possible solely using MODEC since $x_1 \oplus x_2x_3$ is neither monotone in x_1 nor in x_2. If a value function $\mathfrak{I}$ (that is not necessarily an IVF) is derivative dependent, then this has some interesting implications. First, $\mathfrak{I}$ is unbiased and satisfies MODEC. Second, if $\mathfrak{I}$ is weakly chain-rule decomposable (weakly rank preserving), then it is also chain-rule decomposable (rank preserving). Finally, if $\mathfrak{I}$ satisfies DIC and DUM, then it is also bounded by zero and one. As a consequence, if $\mathfrak{I}$ is derivative dependent and satisfies DIC, DUM, and TYPE, then $\mathfrak{I}$ is an IVF.

3.2 Induced Relations

In this section, we will establish foundational relations between IVFs. Recall that f is a *threshold function* if

$$f(\boldsymbol{u}) = 1 \quad \text{iff} \quad \textstyle\sum_{x \in X} w_x \boldsymbol{u}(x) \geq \delta \quad \forall \boldsymbol{u} \in \{0,1\}^X,$$

where $\{w_x\}_{x \in X} \subseteq \mathbb{R}$ is a set of weights and $\delta \in \mathbb{R}$ a threshold.

Theorem 1. *Let $\mathfrak{I}$ be an IVF, $f, g, h \in \mathbb{B}(X)$, $x, y \in X$. Then:*

(1) If f is symmetric, then $\mathfrak{I}_x(f) = \mathfrak{I}_y(f)$.

(2) If $\mathfrak{I}$ is unbiased and f is dual to g, then $\mathfrak{I}_x(f) = \mathfrak{I}_x(g)$.

(3) If f is a threshold function with weights $\{w_x\}_{x \in X} \subseteq \mathbb{R}$, then $|w_x| \geq |w_y|$ implies $\mathfrak{I}_x(f) \geq \mathfrak{I}_y(f)$.

(4) If f is monotonically modular in g and $x \in \mathtt{dep}(g)$, then $\mathfrak{I}_x(g) \geq \mathfrak{I}_x(f)$.

(5) If $\mathfrak{I}$ is derivative dependent and $x \notin \mathtt{dep}(g)$, then $\mathfrak{I}_x(h \oplus g) = \mathfrak{I}_x(h)$.

(6) If $\mathfrak{I}$ is (weakly) chain-rule decomposable, then it is (weakly) rank preserving.

For the case of threshold functions, Theorem 1 shows in (3) that any IVF will rank variables according to their absolute weights. In (4), it is stated that the if a function is monotonically embedded somewhere, the importance of variables in that function can only decrease, e.g., $\mathfrak{I}_x(xy) \geq \mathfrak{I}_x(xyz)$. Moreover, in (5), if derivative dependence is satisfied, $\oplus$-parts without the variable can be dropped. As a consequence, $\mathfrak{I}_x(f) = 1$ whenever f is a parity function and $x \in \mathtt{dep}(f)$.

4 Instances of Importance Value Functions

In this section, we show that IVFs can be instantiated with several notions for importance values from the literature and thus provide a unifying framework.

4.1 Blame

Chockler, Halpern, and Kupferman's (CHK) notions of *responsibility* [Chockler *et al.*, 2008] and *blame* [Chockler and Halpern, 2004] measure the importance of x in f through the number of variables that have to be flipped in an assignment $\boldsymbol{u}$ until x becomes *critical*, i.e., "flipping" x changes the outcome of f to its complement. Towards a formalization, let

$$\mathrm{flip}_S(\boldsymbol{u})(x) = \begin{cases} \overline{\boldsymbol{u}}(x) & \text{if } x \in S \\ \boldsymbol{u}(x) & \text{otherwise} \end{cases}$$

denote the assignment that flips variables in S. We now rely on the following notion of critical set:

Definition 4 (Critical sets)**.** A *critical set* of $x \in X$ in $f \in \mathbb{B}(X)$ under assignment $\boldsymbol{u}$ over X is a set $S \subseteq X \backslash \{x\}$ where

$$f(\boldsymbol{u}) = f(\mathrm{flip}_S(\boldsymbol{u})) \text{ and } f(\boldsymbol{u}) \neq f(\mathrm{flip}_{S \cup \{x\}}(\boldsymbol{u})).$$

We define $\mathrm{scs}_x^{\boldsymbol{u}}(f)$ as the size of the smallest critical set, and set $\mathrm{scs}_x^{\boldsymbol{u}}(f) = \infty$ if there is no such critical set.

Example. The set $S = \{y\}$ is critical for x in $f = x \vee y$ under $\boldsymbol{u} = x/1; y/1$. It is also the smallest critical set. On the other hand, there is no critical set if $\boldsymbol{u} = x/0; y/1$.

The responsibility of x for f under $\boldsymbol{u}$ is inversely related to $\mathrm{scs}_x^{\boldsymbol{u}}(f)$. Using the following notion of a *share function*, we generalize the original notion of responsibility [Chockler *et al.*, 2008]:

Definition 5 (Share function)**.** Call $\rho \colon \mathbb{N} \cup \{\infty\} \to \mathbb{R}$ a *share function* if (i) ρ is monotonically decreasing, (ii) $\rho(\infty) = \lim_{n \to \infty} \rho(n) = 0$, and (iii) $\rho(0) = 1$.

In particular, we consider three instances of share functions:

- $\rho_{\mathrm{exp}}(k) = 1/2^k$,
- $\rho_{\mathrm{frac}}(k) = 1/(k+1)$,
- $\rho_{\mathrm{step}}(k) = 1$ for $k = 0$ and $\rho(k) = 0$ otherwise.

Given a share function ρ, the *responsibility* of x for f under $\boldsymbol{u}$ is defined as $\rho(\mathrm{scs}_x^{\boldsymbol{u}}(f))$. Note that $\rho_{\mathrm{frac}}(\mathrm{scs}_x^{\boldsymbol{u}}(f))$ implements the classical notion of responsibility [Chockler *et al.*, 2008]. While responsibility corresponds to the size of the smallest critical set in a fixed assignment, CHK's *blame* [Chockler *et al.*, 2008] is a global perspective and fits our notion of value function. It is the expected value of the responsibility (we restrict ourselves to uniform distributions):

Definition 6 (Blame)**.** For a share function ρ, we define the ρ-*blame* as value function $\mathbf{B}^\rho$ where for any $x \in X$, $f \in \mathbb{B}(X)$:

$$\mathbf{B}_x^\rho(f) = \mathbb{E}_{\boldsymbol{u} \in \{0,1\}^X}[\rho(\mathrm{scs}_x^{\boldsymbol{u}}(f))].$$

Example. Let $f = x \vee y$. To compute the importance of x we can count the number of times $\mathrm{scs}_x^{\boldsymbol{u}}(f) = 0, 1, 2, \ldots, \infty$ occurs if $\boldsymbol{u}$ ranges over the assignments for $\{x, y\}$: $\mathrm{scs}_x^{\boldsymbol{u}}(f) = \infty$ happens once, $\mathrm{scs}_x^{\boldsymbol{u}}(f) = 0$ happens twice, and $\mathrm{scs}_x^{\boldsymbol{u}}(f) = 1$ occurs once. Thus, $\mathbf{B}_x^\rho(f) = 1/4 \cdot \rho(\infty) + 1/2 \cdot \rho(0) + 1/4 \cdot \rho(1)$, which is $5/8$ for $\rho = \rho_{\mathrm{exp}}$.

Independent of ρ, the blame is always an IVF:

Theorem 2. *$\mathbf{B}^\rho$ is an unbiased IVF for any share function ρ.*

In full generality, the blame violates the optional properties for IVFs (see Section 3.1). For example, if $\rho \neq \rho_{\mathrm{step}}$, then the ρ-blame is neither chain-rule decomposable nor derivative dependent, and one can find counterexamples for the rank-preservation property for ρ_{frac} and ρ_{exp}:

Proposition 1. *Let ρ be a share function. Then the following statements are equivalent:*

(i) *$\mathbf{B}^\rho$ is weakly chain-rule decomposable,*
(ii) *$\mathbf{B}^\rho$ is derivative dependent, and*
(iii) $\rho = \rho_{\mathrm{step}}$.

Further, neither $\mathbf{B}^{\rho_{\mathrm{frac}}}$ nor $\mathbf{B}^{\rho_{\mathrm{exp}}}$ are weakly rank preserving.

To give an example for the reason why the ρ_{frac}-blame is not weakly rank preserving, consider $g = x_1\overline{x}_0\overline{x}_2 \vee \overline{x}_1 x_0 \vee x_3$ and $f = g \vee z$. Note that f is clearly monotonically modular in g – only z is added as fresh variable. Nevertheless, the order of x_0 and x_3 changes:

$$\mathbf{B}_{x_0}^{\rho_{\mathrm{frac}}}(g) = 0.6302 < 0.7188 = \mathbf{B}_{x_3}^{\rho_{\mathrm{frac}}}(g)$$
$$\mathbf{B}_{x_0}^{\rho_{\mathrm{frac}}}(f) = 0.4802 > 0.4688 = \mathbf{B}_{x_3}^{\rho_{\mathrm{frac}}}(f).$$

Intuitively, this is because by CHK's definition of critical sets: for all Boolean functions h, variables x and assignments $\boldsymbol{u}$,

$$h(\boldsymbol{u}) = 1, h_{x/1} \geq h_{x/0}, \boldsymbol{u}(x) = 0 \implies \mathrm{scs}_x^{\boldsymbol{u}}(h) = \infty.$$

Hence, whenever an assignment $\boldsymbol{u}$ satisfies the premise for x in h, the responsibility of x for h under $\boldsymbol{u}$ will be zero.

For x_3, this is more frequently the case in g than in f (19% vs. 34% of all assignments). On the other hand, there is always a critical set for x_0 in both f and g. Partly for this reason, the importance of x_3 decreases more than x_0 when switching from g to f.

Modified Blame

We modify the definition of critical sets in order to derive a *modified blame* that satisfies more optional properties for a wider class of share functions.

For a Boolean function f, an assignment $\boldsymbol{u}$ over X and a variable x, the *modified* scs is defined as the size $\mathrm{mscs}_x^{\boldsymbol{u}}(f)$ of the smallest set $S \subseteq X \setminus \{x\}$ that satisfies

$$f\big(\mathrm{flip}_S(\boldsymbol{u})\big) \neq f\big(\mathrm{flip}_{S\cup\{x\}}(\boldsymbol{u})\big).$$

If there is no such set, we set $\mathrm{mscs}_x^{\boldsymbol{u}}(f) = \infty$.

Example. The condition for critical sets is relaxed, hence $\mathrm{mscs}_x^{\boldsymbol{u}}(f)$ provides a lower bound for $\mathrm{scs}_x^{\boldsymbol{u}}(f)$. Let for example $f = x \vee y$ and $\boldsymbol{u} = x/0; y/1$. Then

$$\mathrm{mscs}_x^{\boldsymbol{u}}(f) = 1 < \infty = \mathrm{scs}_x^{\boldsymbol{u}}(f).$$

The definitions for responsibility and blame are analogous for the modified version, replacing scs by mscs. We denote by $\mathbf{MB}^\rho$ the *modified ρ-blame*, which is (in contrast to $\mathbf{B}^\rho$) always derivative dependent and even chain-rule decomposable if ρ is an exponential- or stepping-function:

Theorem 3. $\mathbf{MB}^\rho$ *is an unbiased, derivative-dependent IVF for any share function* ρ*. If there is* $0 \le \lambda < 1$ *so that* $\rho(k) = \lambda^k$ *for all* $k \ge 1$*, then* $\mathbf{MB}^\rho$ *is chain-rule decomposable.*

4.2 Influence

The influence [Ben-Or and Linial, 1985; Kahn *et al.*, 1988; O'Donnell, 2014] is a popular importance measure, defined as the probability that flipping the variable changes the function's outcome for uniformly distributed assignments:

Definition 7. The *influence* is the value function $\mathbf{I}$ defined by $\mathbf{I}_x(f) = \mathbb{E}[\mathrm{D}_x f]$ for all $f \in \mathbb{B}(X)$ and variables $x \in X$.

It turns out that the influence is a special case of blame:

Proposition 2. $\mathbf{I} = \mathbf{MB}^{\rho_{\mathrm{step}}} = \mathbf{B}^{\rho_{\mathrm{step}}}$.

Since $\rho_{\mathrm{step}}(k) = 0^k$ for $k \ge 1$, Proposition 2 and Theorem 3 show that the influence is a derivative-dependent, rank-preserving, and chain-rule decomposable IVF.

Characterizing the Influence

Call a value function $\mathfrak{I}$ *cofactor-additive* if for all Boolean functions f and variables $x \neq z$:

$$\mathfrak{I}_x(f) = {}^1\!/\!{}_2 \cdot \mathfrak{I}_x(f_{z/0}) + {}^1\!/\!{}_2 \cdot \mathfrak{I}_x(f_{z/1}).$$

Using this notion, we *axiomatically characterize* the influence as follows.

Theorem 4. *A value function* $\mathfrak{I}$ *satisfies* DIC*,* DUM*, and* cofactor-additivity *if and only if* $\mathfrak{I} = \mathbf{I}$.

Remark. A relaxed version of *cofactor-additivity* assumes the existence of $\alpha_z, \beta_z \in \mathbb{R}$ for $z \in X$ such that for all $x \neq z$:

$$\mathfrak{I}_x(f) = \alpha_z \mathfrak{I}_x(f_{z/0}) + \beta_z \mathfrak{I}_x(f_{z/1}).$$

This, together with the assumption that $\mathfrak{I}$ satisfies TYPE, DUM and DIC, implies $\alpha_z = \beta_z = {}^1\!/\!{}_2$. Hence, another characterization of the influence consists of TYPE, DUM, DIC, and *relaxed cofactor-additivity*.

Moreover, we give a *syntactic characterization* of the influence by a comparison to the two-sided Jeroslow-Wang heuristic used for SAT-solving [Jeroslow and Wang, 1990; Hooker and Vinay, 1995; Marques-Silva, 1999]. This value is defined for families of sets of literals, which are sets of subsets of $X \cup \{\overline{z} : z \in X\}$, and it weights subsets that contain x or $\overline{x}$ by their respective lengths:

Definition 8 ([Hooker and Vinay, 1995])**.** Let $\mathcal{D}$ be a family of sets of literals. The *two-sided Jeroslow-Wang value* for a variable x is defined as

$$\mathbf{JW}_x(\mathcal{D}) = \textstyle\sum_{C \in \mathcal{D} \text{ s.t. } x \in C \text{ or } \overline{x} \in C} 2^{-|C|}$$

We call a set C of literals *trivial* if there is a variable x such that $x \in C$ and $\overline{x} \in C$. For a variable x, say that $\mathcal{D}$ is *x-orthogonal* if for all $C, C' \in \mathcal{D}$, $C \neq C'$, there is a literal $\eta \notin \{x, \overline{x}\}$ such that $\eta \in C$ and $\overline{\eta} \in C'$. Orthogonality is well-studied for DNFs [Crama and Hammer, 2011b]. The two-sided Jeroslow-Wang value and the influence agree up to a factor of two for some families of sets of literals when interpreting them as DNFs:

Theorem 5. *Let* $\mathcal{D}$ *be a family of sets of literals such that all of its elements are non-trivial, and let* x *be variable such that* $\mathcal{D}$ *is* x*-orthogonal. Then:*

$$\mathbf{I}_x\big(\textstyle\bigvee_{C \in \mathcal{D}} \bigwedge_{\eta \in C} \eta\big) \;=\; 2 \cdot \mathbf{JW}_x(\mathcal{D}).$$

A simple example that illustrates Theorem 5 would be $\mathcal{D} = \{\{x, y, z\}, \{y, \overline{z}\}\}$. Note that we can interpret $\mathcal{D}$ as a CNF as well, since the influence does not distinguish between a function and its dual (Theorem 1). Note that every Boolean function can be expressed by a family $\mathcal{D}$ that satisfies the conditions of Theorem 5. For this, we construct the canonical DNF corresponding to f and resolve all monomials that differ only in x.

4.3 Cooperative Game Mappings

Attribution schemes analogous to what we call *value functions* were already studied in the context of game theory, most often with emphasis on Shapley- and Banzhaf values [Shapley, 1953; Banzhaf, 1965]. They are studied w.r.t. *cooperative games*, which are a popular way of modeling collaborative behavior. Instead of Boolean assignments, their domains are subsets (coalitions) of X. Specifically, cooperative games are of the form $v\colon 2^X \to \mathbb{R}$, in which the value $v(S)$ is associated with the payoff that variables (players) in S receive when collaborating. Since more cooperation generally means higher payoffs, they are often assumed to be monotonically increasing w.r.t. set inclusion. In its unconstrained form, they are essentially pseudo Boolean functions.

We denote by $\mathbb{G}(X)$ the set of all cooperative games. If $\mathtt{image}(v) \subseteq \{0,1\}$, then we call v *simple*. For a cooperative game v, we denote by $\partial_x v$ the cooperative game that computes the "derivative" of v w.r.t. x, which is $\partial_x v(S) = v(S \cup \{x\}) - v(S \setminus \{x\})$. We compose cooperative games using operations such as $\cdot, +, -, \wedge, \vee$ etc., where $(v \circ w)(S) = v(S) \circ w(S)$. For $\sim\, \in \{\ge, \le, =\}$, we also write $v \sim w$ if $v(S) \sim w(S)$ for all $S \subseteq X$. The set of variables v depends on is defined as $\mathtt{dep}(v) = \{x \in X : \partial_x v \neq 0\}$.

Cooperative game mappings map Boolean functions to cooperative games. Specific *instances* of such mappings have previously been investigated by [Hammer *et al.*, 2000; Biswas and Sarkar, 2021]. We provide a *general definition* of this concept to show how it can be used to construct IVFs.

Definition 9 (CGM)**.** A *cooperative game mapping (CGM)* is a function $\tau\colon \mathbb{B}(X) \to \mathbb{G}(X)$ with $f \mapsto \tau_f$. We call τ *importance inducing* if for all $x, y \in X$, permutations $\sigma\colon X \to X$, and $f, g, h \in \mathbb{B}(X)$:

(BOUND$_{\mathrm{CG}}$) $0 \le \partial_x \tau_f \le 1$.

(DUM$_{\mathrm{CG}}$) $\partial_x \tau_f = 0$ if $x \notin \mathtt{dep}(f)$.

(DIC$_{\text{CG}}$) $\partial_x \tau_x = \partial_x \tau_{\overline{x}} = 1$.

(TYPE$_{\text{CG}}$) (i) $\tau_f(S) = \tau_{\sigma f}(\sigma(S))$ and
(ii) $\tau_f(S) = \tau_{f[y/\overline{y}]}(S)$ for all $S \subseteq X$.

(MODEC$_{\text{CG}}$) $\partial_x \tau_f \geq \partial_x \tau_h$ if
(i) f and h are monotonically modular in g,
(ii) $f_{g/1} \geq h_{g/1}$ and $h_{g/0} \geq f_{g/0}$ and
(iii) $x \in \mathtt{dep}(g)$.

We call τ *unbiased* if $\tau_g = \tau_{\overline{g}}$ for all $g \in \mathbb{B}(X)$.

An example is the *characteristic* CGM ζ given by $\zeta_f(S) = f(\mathbf{1}_S)$, where $\mathbf{1}_S(x) = 1$ iff $x \in S$. We study various importance-inducing CGMs in the following sections. Note that ζ is not importance inducing: for example, it violates BOUND$_{\text{CG}}$ since $\partial_x \zeta_f(\varnothing) = -1$ for $f = \overline{x}$.

The restriction to *importance-inducing* CGMs ensures that compositions with the Banzhaf or Shapley value are valid IVFs (Lemma 2). These CGMs satisfy properties that are related to Definition 2: τ_f should be monotone ($0 \leq \partial_x \tau_f$), irrelevant variables of f are also irrelevant for τ_f (DUM$_{\text{CG}}$), etc. In an analogous fashion, we can think of properties related to Definition 3:

Definition 10. A CGM τ is called

- *chain-rule decomposable*, if for all $f, g \in \mathbb{B}(X)$ such that f is modular in g and $x \in \mathtt{dep}(g)$:
$$\partial_x \tau_f = (\partial_x \tau_g)(\partial_g \tau_f),$$
where $\partial_g \tau_f = \partial_{x_g} \tau_{f[g/x_g]}$ for some $x_g \notin \mathtt{dep}(f)$. We call τ *weakly cain-rule decomposable* if this holds for all cases where f is monotonically modular in g.
- *derivative dependent*, if for all $f, g \in \mathbb{B}(X)$, $x \in X$
$$\mathrm{D}_x f \geq \mathrm{D}_x g \implies \partial_x \tau_f \geq \partial_x \tau_g.$$

Since (weak) rank-preservation for value functions uses an IVF in its premise, it cannot be stated naturally at the level of CGMs. Let us now define the following abstraction, which captures Shapley and Banzhaf values:

Definition 11. Call $\mathfrak{E}\colon X \times \mathbb{G}(X) \to \mathbb{R}, (x, v) \mapsto \mathfrak{E}_x(v)$ a *value function for cooperative games*. Call $\mathfrak{E}$ an *expectation of contributions* if there are weights $c(0), \ldots, c(n{-}1) \in \mathbb{R}$ such that for all $v \in \mathbb{G}(X)$ and $x \in X$:

$$\sum_{S \subseteq X \setminus \{x\}} c(|S|) = 1 \quad \text{and} \quad \mathfrak{E}_x(v) = \sum_{S \subseteq X \setminus \{x\}} c(|S|) \cdot \partial_x v(S).$$

If $\mathfrak{E}$ is an expectation of contributions, then $\mathfrak{E}_x(v)$ is indeed the expected value of $\partial_x v(S)$ in which every $S \subseteq X \setminus \{x\}$ has probability $c(|S|)$. The *Banzhaf* and *Shapley values* are defined as the expectations of contributions with weights:

$$c_{\mathrm{Bz}}(k) = \tfrac{1}{2^{n-1}}\ (\mathbf{Bz}) \qquad \text{and} \qquad c_{\mathrm{Sh}}(k) = \tfrac{1}{n}\tbinom{n-1}{k}^{-1}\ (\mathbf{Sh}).$$

Observe that there are $\binom{n-1}{k}$ sets of size $k \in \{0, \ldots, n{-}1\}$, so the weights of the Shapley value indeed sum up to one.

If τ is a CGM, then its composition with $\mathfrak{E}$ yields $(\mathfrak{E} \circ \tau)_x(f) = \mathfrak{E}_x(\tau_f)$, which is a value function for Boolean functions. Then every composition with an expectation of contributions is an IVF if the CGM is importance inducing:

Lemma 2. *If τ is an importance-inducing CGM and $\mathfrak{E}$ an expectation of contributions, then $\mathfrak{E} \circ \tau$ is an IVF. If τ is unbiased/derivative dependent, then so is $\mathfrak{E} \circ \tau$. Finally, if τ is (weakly) chain-rule decomposable, then so is $\mathbf{Bz} \circ \tau$.*

In the following sections, we study two novel and the already-known CGM of [Hammer *et al.*, 2000]. By Lemma 2 we can focus on their properties as CGMs, knowing that any composition with the Shapley value or other expectations of contributions will induce IVFs.

Simple Satisfiability-Biased Cooperative Game Mappings

The first CGM interprets the "power" of a coalition as its ability to force a function's outcome to one: If there is an assignment for a set of variables that yields outcome one *no matter* the values of other variables, we assign this set a value of one, and zero otherwise.

Definition 12. The *dominating CGM* ω is defined as

$$\omega_f(S) = \begin{cases} 1 & \text{if } \exists \boldsymbol{u} \in \{0,1\}^S.\ \forall \boldsymbol{w} \in \{0,1\}^{X \setminus S}.\ f(\boldsymbol{u}; \boldsymbol{w}). \\ 0 & \text{otherwise.} \end{cases}$$

Example. Let $f = x \vee (y \oplus z)$. We have $\omega_f(\{y, z\}) = 1$ since $f_{\boldsymbol{u}} = 1$ for $\boldsymbol{u} = y/1; z/0$. On the other hand, $\omega_f(\{y\}) = 0$, since $x/0; z/1$ resp. $x/0; z/0$ falsify $f_{y/1}$ and $f_{y/0}$.

Theorem 6. *The dominating CGM is weakly chain-rule decomposable and importance inducing.*

Example. Let $\mathbf{Z}$ be the expectation of contributions with $c(0) = 1$, i.e., $\mathbf{Z}_x(v) = v(\{x\}) - v(\varnothing)$. By Lemma 2 and Theorem 6, the mapping

$$(\mathbf{Z} \circ \omega)_x(f) = \begin{cases} 1 & \text{if } f \neq 1 \text{ and } f_{x/0} = 1 \text{ or } f_{x/1} = 1 \\ 0 & \text{otherwise} \end{cases}$$

is an IVF. Intuitively, x has the highest importance if the function is falsifiable and there is a setting for x that forces the function to one. Otherwise, x has an importance of zero.

Biasedness and rank preservation. The dominating CGM is biased: Consider $g = x \vee (y \oplus z)$ with $\overline{g} = \overline{x} \wedge (\overline{y} \oplus z)$. Note that $\omega_g(S) = 1$ for $S = \{x\}$ while $\omega_{\overline{g}}(S) = 0$, which shows biasedness. Composing ω with the Banzhaf value yields

$$\begin{aligned} (\mathbf{Bz} \circ \omega)_{(\cdot)}(g):& \quad z : 0.25 \ = \ y : 0.25 \ < \ x : 0.75, \\ (\mathbf{Bz} \circ \omega)_{(\cdot)}(\overline{g}):& \quad z : 0.25 \ = \ y : 0.25 \ = \ x : 0.25, \end{aligned}$$

One can force g to one by controlling either x or *both* y and z, so x is rated higher than the others. But to force $\overline{g}$ to one, control over all variables is required, so all variables in $\overline{g}$ have the same importance.

Since g is modular in $\overline{g}$, we also obtain a counterexample for rank preservation:

$$\begin{aligned} & (\mathbf{Bz} \circ \omega)_y(\overline{g}) \geq (\mathbf{Bz} \circ \omega)_x(\overline{g}) \\ \text{does not imply} \quad & (\mathbf{Bz} \circ \omega)_y(g) \geq (\mathbf{Bz} \circ \omega)_x(g). \end{aligned}$$

However, *weak* rank preservation is fulfilled by $\mathbf{Bz} \circ \omega$ since it is weakly chain-rule decomposable by Theorem 6 and Lemma 2. Then the claim follows with Theorem 1.

A dual to the dominating CGM. One can think of a dual notion of the CGM ω that reverses the order of both quantifiers. Intuitively, we are now allowed to choose an assignment *depending* on the values of the remaining variables:

Definition 13. The *rectifying CGM* ν is defined as

$$\nu_f(S) = \begin{cases} 1 & \text{if } \forall \boldsymbol{w} \in \{0,1\}^{X \setminus S}.\ \exists \boldsymbol{u} \in \{0,1\}^S.\ f(\boldsymbol{u};\boldsymbol{w}). \\ 0 & \text{otherwise.} \end{cases}$$

If we compose ν with an expectation of contributions that satisfies $c(k) = c(n{-}1{-}k)$ for all $k \in \{0,\dots,n{-}1\}$, which is a condition satisfied both by the Shapley and Banzhaf values, the induced importance of a variable equals its importance w.r.t. ω and the negated function:

Proposition 3. *Let $\mathfrak{E}$ be an expectation of contributions with $c(k) = c(n{-}1{-}k)$ for all $k \in \{0,\dots,n{-}1\}$. Then for all $g \in \mathbb{B}(X)$ and $x \in X$:*

$$(\mathfrak{E} \circ \omega)_x(g) = (\mathfrak{E} \circ \nu)_x(\overline{g})$$

We now discuss connections to the influence. If a Boolean function is monotone, and we "control" a set of variables S, the best towards satisfaction (resp. falsification) is to set all variables in S to one (resp. to zero). This can be used to show that both $\mathbf{Bz} \circ \omega$ and $\mathbf{Bz} \circ \nu$ agree with the influence:

Proposition 4. *Let f be a monotone Boolean function and x a variable. Then $(\mathbf{Bz} \circ \omega)_x(f) = (\mathbf{Bz} \circ \nu)_x(f) = \mathbf{I}_x(f)$.*

A Constancy-Based Cooperative Game Mapping

Hammer, Kogan and Rothblum [Hammer *et al.*, 2000] (HKR) defined a CGM that measures the power of variables by how constant they make a function if assigned random values. It depends on the following notion of *constancy measure*:

Definition 14. We call a mapping $\kappa \colon [0,1] \to [0,1]$ a *constancy measure* if (i) κ is convex, (ii) $\kappa(0) = 1$, (iii) $\kappa(x) = \kappa(1{-}x)$, and (iv) $\kappa(1/2) = 0$.

The following functions are instances of constancy measures:

- $\kappa_{\text{quad}}(a) = 4(a - 1/2)^2$,
- $\kappa_{\log}(a) = 1 + a\text{lb}(a) + (1{-}a)\text{lb}(1{-}a)$ with $0\text{lb}(0) = 0$,
- $\kappa_{\text{abs}}(a) = 2|a - 1/2|$.

For a constancy measure κ and a Boolean function f, the *κ-constancy of f* is the value $\kappa(\mathbb{E}[f])$, which measures how balanced the share of ones and zeros is. It is close to one if f is very unbalanced and close to zero if the share of zeros and ones in f is (almost) the same. The power of a set of variables S is now measured in terms of the expected κ-constancy of f if variables in S are fixed to random values:

Definition 15 ([Hammer *et al.*, 2000])**.** Given a constancy measure κ, we define the CGM H^κ by

$$\mathrm{H}^\kappa_f(S) = \mathbb{E}_{\boldsymbol{a} \in \{0,1\}^S}[\kappa(\mathbb{E}[f_{\boldsymbol{a}}])].$$

Example. Let $f = x \vee y \vee z$ and $S = \{x\}$. We obtain $\mathrm{H}^\kappa_f(S) = 1/2 \cdot \kappa(3/4) + 1/2 \cdot \kappa(1)$, since

$$\mathbb{E}[f_{x/0}] = 3/4 \quad \text{and} \quad \mathbb{E}[f_{x/1}] = 1.$$

Setting x to zero does not determine f completely, while setting it to one also sets f to one, i.e., makes it constant. The measure then gives a lower value to the less-constant cofactor, a higher value to the more-constant cofactor and computes the average. For this example and $\kappa = \kappa_{\text{abs}}$, we obtain $\mathrm{H}^\kappa_f(S) = 3/4$ due to $\kappa(3/4) = 1/2$ and $\kappa(1) = 1$.

Theorem 7 shows that $\mathrm{H}^{\kappa_{\text{quad}}}$ is a chain-rule decomposable and importance-inducing CGM. It is open whether other constancy measures are importance inducing too.

Theorem 7. *Suppose κ is a constancy measure. Then H^κ is an unbiased CGM that satisfies* $\textsc{Bound}_{\text{CG}}$*,* $\textsc{Dic}_{\text{CG}}$*,* $\textsc{Dum}_{\text{CG}}$*, and* $\textsc{Type}_{\text{CG}}$*. Further, $\mathrm{H}^{\kappa_{\text{quad}}}$ is chain-rule decomposable and satisfies* $\textsc{ModEC}_{\text{CG}}$*.*

Example. For the special case where $\kappa = \kappa_{\text{quad}}$, note that

$$1/2 \cdot \kappa(a) + 1/2 \cdot \kappa(b) - \kappa(1/2 \cdot a + 1/2 \cdot b) = (a-b)^2.$$

Using $\mathbb{E}[f] = 1/2 \cdot \mathbb{E}[f_{x/1}] + 1/2 \cdot \mathbb{E}[f_{x/0}]$, this implies

$$(\mathbf{Z} \circ \mathrm{H}^\kappa)_x(f) = (\mathbb{E}[f_{x/1}] - \mathbb{E}[f_{x/0}])^2,$$

where $\mathbf{Z}$ is again the expectation of contributions with

$$\mathbf{Z}_x(v) = v(\{x\}) - v(\varnothing).$$

The value $\mathbf{Z} \circ \mathrm{H}^\kappa$ is an IVF according Lemma 2 and Theorem 7. In contrast to derivative-dependent IVFs, $\mathbf{Z} \circ \mathrm{H}^\kappa$ assigns low values to variables in parity-functions: for $f = x \oplus y$, we have $\mathbb{E}[f_{x/1}] = \mathbb{E}[f_{x/0}]$, and thus $(\mathbf{Z} \circ \mathrm{H}^\kappa)_x(f) = 0$.

Derivative dependence. This property cannot be achieved, as witnessed by $f = x \oplus y$ and $g = x$. Due to $\mathrm{D}_x f = \mathrm{D}_x g$, it suffices to show that $\partial_x \mathrm{H}^\kappa_f \neq \partial_x \mathrm{H}^\kappa_g$ holds for all κ. Note that

$$\mathbb{E}[f_{x/0}] = 1/2,\ \mathbb{E}[f_{x/1}] = 1/2,\ \mathbb{E}[g_{x/0}] = 0,\ \mathbb{E}[g_{x/1}] = 1,$$

and $\mathbb{E}[f] = \mathbb{E}[g] = 1/2$. Thus, for all constancy measures κ,

$$\partial_x \mathrm{H}^\kappa_f(\varnothing) = 1/2 \cdot \kappa(1/2) + 1/2 \cdot \kappa(1/2) - \kappa(1/2) = 0,$$
$$\partial_x \mathrm{H}^\kappa_g(\varnothing) = 1/2 \cdot \kappa(1) + 1/2 \cdot \kappa(0) - \kappa(1/2) = 1,$$

which shows $\partial_x \mathrm{H}^\kappa_f \neq \partial_x \mathrm{H}^\kappa_g$.

5 Computing Importance Values

In this section, we present and evaluate computation schemes for blame, influence, and CGMs. While there exists a practical approach based on model counting for the influence in CNFs [Traxler, 2009], we are only aware of naïve computations of CHK's blame [Dubslaff *et al.*, 2022].

Blame. We focus on the modified blame. CHK's blame can be computed in a very similar fashion. Observe that for a Boolean function f and $x \in X$,

$$\mathbf{MB}^\rho_x(f) = \mathbb{E}[\gamma_0] + \textstyle\sum_{k=1}^{n-1} \rho(k)(\mathbb{E}[\gamma_k] - \mathbb{E}[\gamma_{k-1}]),$$

where γ_k is the Boolean function for which $\gamma_k(\boldsymbol{u}) = 1$ iff $\text{mscs}^{\boldsymbol{u}}_x(f) \leq k$. We devise two approaches for computing $\mathbb{E}[\gamma_k]$. The first represents γ_k through BDDs using the following recursion scheme: $\text{mscs}^{\boldsymbol{u}}_x(f) \leq k$ holds iff

- $k = 0$ and $f(\boldsymbol{u}) \neq f(\text{flip}_{\{x\}}(\boldsymbol{u}))$, or
- $k > 0$ and
  - $\text{mscs}^{\boldsymbol{u}}_x(f) \leq k{-}1$ or
  - there is $y \neq x$ such that $\text{mscs}^{\boldsymbol{u}}_x(f[y/\overline{y}]) \leq k{-}1$.

This allows us to construct BDDs for γ_k from γ_{k-1}, which lends itself to BDD-based approaches since γ_k does not necessarily increase in size as k grows. The second approach introduces new existentially quantified variables in the input formula of f to model occurrences of variables in critical sets of $\text{mscs}^{\boldsymbol{u}}_x(f)$. With an additional cardinality constraint restricting the number of variables in critical sets to at most k, we can use projected model counting to compute $\mathbb{E}[\gamma_k]$.

			(projected) model counting approaches			BDD-based approaches			
Instance	#Variables	#Clauses	Influence (CNF)	Influence (formula)	Blame	Construction	Influence	DCGM	Blame
b02	26	66	5 ms	49 ms	timeout	1 ms	<1 ms	2 ms	3'649 ms
b06	44	122	7 ms	99 ms	timeout	3 ms	<1 ms	6 ms	697'573 ms
b01	45	120	7 ms	110 ms	timeout	4 ms	<1 ms	8 ms	3'068'667 ms
b03	156	376	11 ms	442 ms	timeout	53'934 ms	24 ms	1'776 ms	timeout
b13	352	847	34 ms	1'088 ms	timeout	timeout	timeout	timeout	timeout
b12	1'072	2'911	230 ms	8'555 ms	timeout	timeout	timeout	timeout	timeout

Table 1: Computation time for instances of the ISCAS'99 dataset, timeout set to one hour. BDD columns *Influence*, *DCGM* (construction of the BDD for the dominating CGM), and *Blame* are *without* the BDD construction time for the initial CNF (cf. column *Construction*).

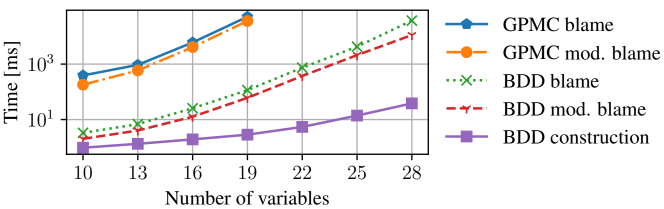

Figure 1: Computation of blame values on random $(n, 3n, 7)$-CNFs (number of variables, number of clauses, clause width). BDD times *include* construction time of the BDD for the initial CNF.

Influence. In case f is given as a CNF formula, we use Traxler's method to compute the influence [Traxler, 2009]. For all other formulas, note that standard satisfiability-preserving transformations do not preserve influence values: For example, applying the Tseytin transformation to $x \vee \overline{x}y$ results in a CNF where x has a higher influence than y.

However, the influence is proportional to the number of models of $\mathrm{D}_x f$. If f is given by a BDD, computing a representation of $\mathrm{D}_x f$ means squaring f's size in the worst case, while the formula-based representation only doubles it. For the latter case, we can count the models of $\mathrm{D}_x f$ using a Tseytin transformation and a standard model counter.

BDD representations of satisfiability-biased CGMs. The dominating CGM computes a simple game, which is essentially a Boolean function, and therefore permits a representation by BDDs. Moreover, using a BDD representation of f, we compute ω_f using a recursion on cofactors of variables z,

$$(\omega_f)_{z/1} = \omega_{f_{z/1}} \vee \omega_{f_{z/0}} \quad \text{and} \quad (\omega_f)_{z/0} = \omega_{f_{z/0} \wedge f_{z/1}}.$$

The Banzhaf value of x in ω_f is then just

$$\mathbb{E}[(\omega_f)_{x/1}] - \mathbb{E}[(\omega_f)_{x/0}],$$

which poses no effort once the BDD of ω_f is constructed. The rectifying CGM can be computed analogously.

Implementation and evaluation. We have implemented Traxler's method and our new computation schemes in Python, using BuDDy [Lind-Nielsen, 1999] as BDD backend with automatic reordering and GPMC [Suzuki *et al.*, 2015; Suzuki *et al.*, 2017] for (projected) model counting. To evaluate our approaches, we conducted experiments on Boolean functions given as CNFs that were either randomly generated or generated from the ISCAS'99 dataset [Davidson, 1999; Compile! Project, 2023]. We always computed importance values w.r.t. the first variable in the input CNF and averaged the timings over 20 runs each. Our experiments were carried out on a Linux system with an i5-10400F CPU at 2.90GHz and 16GB of RAM. To compare our BDD-based and model counting approaches, Figure 1 shows timings for blame computations on random CNFs. Here, the BDD-based approach clearly outperforms the one based on projected model counting. This is also reflected in real-world benchmarks from ISCAS'99 shown in Table 1, where the approach based on model counting runs into timeouts for even small instances. Table 1 shows that computations for influence values based on model counting scale better than the BDD-based approach, mainly due to an expensive initial BDD construction. Computing the BDD of the dominating CGM is done without much overhead once the BDD for the CNF is given.

6 Conclusion

This paper introduced IVFs as a way to formally reason about importance of variables in Boolean functions. We established general statements about IVFs, also providing insights on notions of importance from the literature by showing that they all belong to the class of IVFs. Apart from revealing several relations between known IVFs, we have shown how to generate new ones inspired by cooperative game theory.

For future work, we will study properties with strict importance inequalities, IVFs for sets of variables, IVFs for pseudo Boolean functions, and global values similar to the *total influence* [O'Donnell, 2014]. On the empirical side, the generation of splitting rules for SAT-solvers and variable-order heuristics for BDDs based on different instances of IVFs are promising avenues to pursue.

Acknowledgments

The authors were partly supported by the DFG through the DFG grant 389792660 as part of TRR 248 and the Cluster of Excellence EXC 2050/1 (CeTI, project ID 390696704, as part of Germany's Excellence Strategy) and "SAIL: SustAInable Life-cycle of Intelligent Socio-Technical Systems" (Grant ID NW21-059D), funded by the program "Netzwerke 2021" of the Ministry of Culture and Science of the State of North Rhine-Westphalia, Germany.

References

[Ashenhurst, 1957] Robert Ashenhurst. The Decomposition of Switching Functions. In *Proceedings of an International Symposium on the Theory of Switching*, April 1957.

[Aziz *et al.*, 2015] Rehan Abdul Aziz, Geoffrey Chu, Christian Muise, and Peter Stuckey. #∃SAT: Projected model counting. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 121–137. Springer, 2015.

[Banzhaf, 1965] John Francis Banzhaf. Weighted Voting Doesn't Work: A Mathematical Analysis. *Rutgers Law Review*, 19:317–343, 1965.

[Bartlett and Andrews, 2001] L. M. Bartlett and J. D. Andrews. Comparison of two new approaches to variable ordering for binary decision diagrams. *Quality and Reliability Engineering International*, 17(3):151–158, May 2001.

[Ben-Or and Linial, 1985] Michael Ben-Or and Nathan Linial. Collective Coin Flipping, Robust Voting Schemes and Minima of Banzhaf Values. In *26th Annual Symposium on Foundations of Computer Science (Sfcs 1985)*, pages 408–416, Portland, OR, USA, 1985. IEEE.

[Bioch, 2010] Jan C. Bioch. Decomposition of Boolean Functions. In Yves Crama and Peter L. Hammer, editors, *Boolean Models and Methods in Mathematics, Computer Science, and Engineering*, pages 39–76. Cambridge University Press, first edition, June 2010.

[Birnbaum and Esary, 1965] Zygmunt Wilhelm Birnbaum Birnbaum and J. D. Esary. Modules of Coherent Binary Systems. *Journal of the Society for Industrial and Applied Mathematics*, 13(2):444–462, June 1965.

[Biswas and Sarkar, 2021] Aniruddha Biswas and Palash Sarkar. Influence of a set of variables on a boolean function. *Electron. Colloquium Comput. Complex.*, TR21-111, 2021.

[Bryant, 1992] Randal E. Bryant. Symbolic Boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys*, 24(3):293–318, September 1992.

[Chockler and Halpern, 2004] Hana Chockler and Joseph Y. Halpern. Responsibility and Blame: A Structural-Model Approach. *Journal of Artificial Intelligence Research*, 22:93–115, October 2004.

[Chockler *et al.*, 2008] Hana Chockler, Joseph Y. Halpern, and Orna Kupferman. What Causes a System to Satisfy a Specification? *ACM Transactions on Computational Logic*, 9(3):1–26, June 2008.

[Compile! Project, 2023] Compile! Project. Benchmarks. https://www.cril.univ-artois.fr/kc/benchmarks.html, 2023. Accessed: 2023-01-11.

[Crama and Hammer, 2011a] Yves Crama and Peter L. Hammer. *Boolean Functions: Theory, Algorithms, and Applications*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2011.

[Crama and Hammer, 2011b] Yves Crama and Peter L. Hammer. Orthogonal forms and shellability. In *Boolean Functions: Theory, Algorithms, and Applications*, Encyclopedia of Mathematics and Its Applications, chapter 7, pages 326–350. Cambridge University Press, Cambridge, 2011.

[Davidson, 1999] Scott Davidson. Itc'99 benchmark circuits-preliminary results. In *International Test Conference 1999. Proceedings (IEEE Cat. No. 99CH37034)*, pages 1125–1125. IEEE, 1999.

[Deng and Papadimitriou, 1994] Xiaotie Deng and Christos H. Papadimitriou. On the complexity of cooperative solution concepts. *Mathematics of Operations Research*, 19(2):257–266, 1994.

[Dubslaff *et al.*, 2022] Clemens Dubslaff, Kallistos Weis, Christel Baier, and Sven Apel. Causality in configurable software systems. In *Proceedings of the 44th International Conference on Software Engineering*, ICSE '22, page 325–337, New York, NY, USA, 2022. Association for Computing Machinery.

[Faigle and Kern, 1992] Ulrich Faigle and Walter Kern. The shapley value for cooperative games under precedence constraints. *International Journal of Game Theory*, 21(3):249–266, 1992.

[Golomb, 1959] Solomon Golomb. On the classification of Boolean functions. *IRE Transactions on Information Theory*, 5(5):176–186, May 1959.

[Hammer *et al.*, 2000] Peter L. Hammer, Alexander Kogan, and Uriel G. Rothblum. Evaluation, Strength, and Relevance of Variables of Boolean Functions. *SIAM Journal on Discrete Mathematics*, 13(3):302–312, January 2000.

[Hooker and Vinay, 1995] John N. Hooker and V. Vinay. Branching rules for satisfiability. *Journal of Automated Reasoning*, 15(3):359–383, 1995.

[Jeroslow and Wang, 1990] Robert G. Jeroslow and Jinchang Wang. Solving propositional satisfiability problems. *Annals of Mathematics and Artificial Intelligence*, 1(1-4):167–187, September 1990.

[Kahn *et al.*, 1988] Jeff Kahn, Gil Kalai, and Nathan Linial. The Influence of Variables on Boolean Functions. In *[Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science*, pages 68–80, October 1988.

[Lind-Nielsen, 1999] Jørn Lind-Nielsen. BuDDy: A binary decision diagram package. *Department of Information Technology, Technical University of Denmark*, 1999.

[Marques-Silva, 1999] João Marques-Silva. The Impact of Branching Heuristics in Propositional Satisfiability Algorithms. In G. Goos, J. Hartmanis, J. van Leeuwen, Pedro Barahona, and José J. Alferes, editors, *Progress in Artificial Intelligence*, volume 1695, pages 62–74. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[O'Donnell, 2014] Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

[Shapley, 1953] Loyd. S. Shapley. A Value for n-Person Games. In Harold William Kuhn and Albert William Tucker, editors, *Contributions to the Theory of Games II*, pages 307–318. Princeton University Press, December 1953.

[Shapley, 1967] Lloyd S. Shapley. Compound simple games, III: On committees. Technical report, RAND CORP SANTA MONICA CA, 1967.

[Slepian, 1953] David Slepian. On The Number of Symmetry Types of Boolean Functions of n Variables. *Canadian Journal of Mathematics*, 5:185–193, 1953.

[Suzuki *et al.*, 2015] Ryosuke Suzuki, Kenji Hashimoto, and Masahiko Sakai. An Extension of a DPLL-Based Model-Counting Solver for Projected Model. *JSAI Technical Report*, SIG-FPAI-97-B404:59–64, March 2015.

[Suzuki *et al.*, 2017] Ryosuke Suzuki, Kenji Hashimoto, and Masahiko Sakai. Improvement of Projected Model-Counting Solver with Component Decomposition Using SAT Solving in Components. *JSAI Technical Report*, SIG-FPAI-103-B506:31–36, March 2017.

[Traxler, 2009] Patrick Traxler. Variable influences in conjunctive normal forms. In *International Conference on Theory and Applications of Satisfiability Testing*, pages 101–113. Springer, 2009.