Approximation des explications abductives de taille minimale : application aux arbres de décision binaires

Louenas Bounia*

LIPN, UMR-CNRS 7030, Université Sorbonne Paris-Nord, 93430 Villetaneuse*

Résumé. Dans ce travail, nous traitons le problème de l'approximation des explications abductives de taille minimale pour les arbres de décision en utilisant l'optimisation super-modulaire. La recherche d'une explication abductive de taille minimale est une tâche très coûteuse en temps même pour des classifieurs restreints comme les arbres de décision. Ce problème, étant NP**-difficile** pour les arbres de décisions, rend les approches exactes particulièrement chronophages, surtout pour les instances complexes et les données de grande dimension. Pour surmonter ces défis, des méthodes approximatives ou heuristiques sont souvent préférées, permettant de réduire la charge computationnelle et d'obtenir des résultats plus rapidement, bien que parfois au détriment de l'optimalité. Nous proposons ici un algorithme glouton pour obtenir des approximations efficaces d'explications abductives de taille minimale. Nos expériences montrent que, pour les instances difficiles à expliquer, cet algorithme offre une alternative efficace aux méthodes exactes basées sur les encodages SAT.

1 Introduction

Fournir une explication à une personne consiste à lui donner les informations nécessaires pour comprendre les raisons derrière une décision, ce qui est particulièrement crucial lorsque celle-ci est générée par des modèles d'apprentissage automatique (ML) tels que les arbres de décision, les forêts aléatoires (Breiman, 2001), les machines à vecteurs de support, ou les réseaux de neurones profonds (Miller, 2019; Molnar, 2019). Avec la montée en puissance des applications utilisant ces techniques, la recherche sur l'intelligence artificielle explicable (XAI) est devenue de plus en plus importante. La plupart des méthodes XAI sont des approches indépendantes, dites « agnostiques », avec des méthodes populaires comme LIME (Ribeiro et al., 2016), SHAP (Lundberg et Lee, 2017), et ANCHORS (Ribeiro et al., 2018). Cependant, ces méthodes présentent des limitations en termes de fiabilité. Par exemple, Ignatiev et al. (2019) ont montré qu'une même explication peut être compatible avec plusieurs classes prédites, ce qui pose un problème dans des domaines sensibles comme la médecine, la sécurité informatique, ou même la finance. Cela renforce l'intérêt pour des méthodes offrant des garanties mathématiques plus solides, telles que les méthodes formelles, qui corrigent cette faiblesse et permettent de fournir des explications valides.

Dans cet article, nous nous concentrons sur la recherche d'explications de taille minimales, afin d'expliquer le classement d'une instance d'entrée $\boldsymbol{x}$, étant donné un classifieur binaire.

Nous mettons l'accent sur les explications « abductives », qui répondent à la question « pourquoi ce classement ? ». Bien qu'il n'existe pas de définition formelle de l'interprétabilité (Lipton, 2018), les arbres de décision (Breiman, 2001; Quinlan, 1986) sont largement considérés comme l'un des modèles les plus interprétables pour la classification. Ils sont souvent utilisés pour distiller un modèle opaque en un modèle compréhensible (Frosst et Hinton, 2017) et servent de base à des modèles plus complexes, comme les forêts aléatoires (RF) (Breiman, 2001) et les arbres boostés (XGBOOST) (Chen et Guestrin, 2016). L'interprétabilité des arbres de décision repose sur leur transparence, chaque nœud ayant une signification claire, mais aussi sur leur capacité à fournir des explications locales. À partir de l'instance à classer, il est facile d'extraire une explication abductive, appelée « raison directe » (Audemard et al., 2022; Louenas, 2023). Cependant, les raisons directes peuvent contenir des attributs redondants (Izza et al., 2020), ce qui justifie l'intérêt pour des explications non redondantes, telles que les raisons suffisantes (Darwiche et Hirth, 2020) (ou PI-explications (Shih et al., 2018)) ou celles de taille minimales.

En identifiant le plus petit sous-ensemble d'attributs suffisant pour justifier une décision, on améliore l'intelligibilité et la transparence du modèle, tout en respectant les limites cognitives des utilisateurs. Cependant, le calcul d'une explication abductive de taille minimale est généralement très complexe. Par exemple, le problème de trouver une raison suffisante de taille minimale pour les forêts aléatoires est DP-complet (Izza et Marques-Silva, 2021), et NP-difficile pour les arbres de décision (Audemard et al., 2022). Par conséquent, cet article explore l'approximation de ces explications lorsque le classifieur est un arbre de décision.

Contribution. Dans ce travail, nous nous intéressons à l'approximation des raisons suffisantes de taille minimale pour une instance $\boldsymbol{x}$ étant donné un classifieur h représenté par un arbre de décision T. Bien que dériver des explications à partir d'un arbre de décision soit computationnellement faisable (Audemard et al., 2021), lorsque l'arbre ou l'instance d'entrée est de grande dimension, le calcul d'une raison suffisante de taille minimale peut devenir inabordable en termes de temps de calcul (Louenas, 2023), en raison de la complexité de ce problème qui est NP-difficile. Pour répondre à ce défi, nous formulons le problème du calcul des explications abductives de taille minimale comme un problème de recherche solution de taille minimale de la fonction d'erreur d'explication qui atteint une borne d'erreur α. Nous proposons également un algorithme glouton pour calculer des approximations efficaces de ce problème, offrant des garanties mathématiques sur la taille de la solution produite. Nous comparons empiriquement nos approches à deux encodages SAT de référence (Audemard et al., 2022; Arenas et al., 2022). Nos résultats démontrent l'efficacité de notre méthode pour approcher la raison suffisante de taille minimale, en particulier pour les instances difficiles, tout en soulignant la performance empirique de nos algorithmes.

2 Arbre de décision, Explications abductives

Préliminaires. Pour un entier n, soit $[n]$ l'ensemble $\{1, \ldots, n\}$. On note $\mathcal{F}_n$ la classe de toutes les fonctions booléennes de $\{0,1\}^n$ à $\{0,1\}$, $X_n = \{x_1, \ldots, x_n\}$ désigne l'ensemble des variables booléennes. Toute affectation $\boldsymbol{x} \in \{0,1\}^n$ est appelée une *instance*. Un *littéral* ℓ est une variable x_i ou sa négation $\overline{x}_i$. Un *terme* t est une conjonction de littéraux [1], et une

1. Nous précisons dans ce travail qu'un terme peut être vue comme un ensemble de littéraux.

clause c est une disjonction de littéraux. Une formule DNF est une disjonction de termes et une formule CNF est une conjonction de clauses. Une formule f est *consistante* si et seulement si elle a un modèle [2]. Étant donné une instance $z \in \{0,1\}^n$, le terme correspondant est défini comme $t_z = \bigwedge_{i=1}^{n} x_i^{z_i} = \{x_1^{z_1}, \cdots, x_n^{z_n}\}$ où $x_i^0 = \overline{x}_i$ et $x_i^1 = x_i$. Un impliquant d'une fonction booléenne f est un terme qui implique f. Un impliquant premier de f est un impliquant t de f tel qu'aucun sous-ensemble de t n'est un impliquant de f. Une instance partielle est un vecteur $\boldsymbol{z} \in \{0,1,*\}^n$, où $z_i = *$ indique que la i-ème caractéristique de $\boldsymbol{z}$ est indéfinie. Une instance $\boldsymbol{x}$ est couverte par $\boldsymbol{z}$ si $x_i = z_i$ pour toutes les caractéristiques $i \in [n]$ telles que $z_i \neq *$. Pour un sous-ensemble S de caractéristiques, la restriction de $\boldsymbol{x}$ à S, notée x_S, est l'instance partielle dans $\{0,1,*\}^n$ telle que, pour chaque $i \in [n]$, $(x_S)_i = x_i$ si $i \in S$, et $(x_S)_i = *$ sinon. Toute instance $y \in \{0,1\}^n$ est couverte par x_S si et seulement si $y_S = x_S$.

2.1 Arbre de décision

Arbre de décision binaire. Un arbre de décision binaire sur X_n est un arbre binaire T, dont chacun des nœuds internes est étiqueté avec l'une des n variables booléennes d'entrée de X_n, et dont les feuilles sont étiquetées par 0 ou 1. Il est supposé que chaque variable apparaît au plus une fois sur n'importe quel chemin racine-feuille (propriété de read-one). La valeur $T(\boldsymbol{x}) \in \{0,1\}$ attribuée par T à une instance $\boldsymbol{x}$ est déterminée par l'étiquette de la feuille atteinte à partir de la racine, en suivant le chemin cohérent avec les caractéristiques de $\boldsymbol{x}$. La taille de T, notée $|T|$, est donnée par le nombre de ses nœuds. La classe des arbres de décision est notée $\mathtt{DT}_n$. Il est bien connu que tout arbre de décision $T \in \mathtt{DT}_n$ peut être transformé en temps linéaire en une disjonction de termes équivalente, notée $\mathtt{DNF}(T)$. Cette DNF est orthogonale, dans laquelle chaque terme correspond à un chemin de la racine à une feuille étiquetée 1. T peut être transformé en une conjonction de clauses, notée $\mathtt{CNF}(T)$ (Audemard et al., 2022).

Définition 1 (DNF orthogonale)**.** *Une* DNF $\phi = \{t_1, \cdots, t_m\}$ *est orthogonale si et seulement si* $\forall i, j \in [m]$*, si* $i \neq j$ *alors* $t_i \cup t_j$ *est inconsistant.*

La définition 1 (Crama et Hammer, 2011) stipule que les termes t_i et t_j sont mutuellement exclusifs, c'est-à-dire qu'ils ne partagent aucun modèle commun où ils sont tous deux vrais. Grâce à la structure spécifique des DNF orthogonales, le comptage des modèles peut se faire en temps linéaire (en $O(m)$). Le nombre total de modèles est donné par : $w(\phi) = \sum_{k=1}^{m} 2^{n-|t_k|}$. Pour un arbre $T \in \mathtt{DT}_n$, une instance $\boldsymbol{x} \in \{0,1\}^n$ et un ensemble S d'attributs, soit t_{x_S} le terme associé à l'instance partielle x_S, c'est-à-dire : $t_{x_S} = \bigcup_{i=1}^{n} (\{x_i : (x_S)_i = 1\} \cup \{\overline{x_i} : (x_S)_i = 0\})$. Par construction, le conditionnement $\mathtt{DNF}(T) \mid t_{x_S}$ est une DNF orthogonale (Darwiche, 1999) et en conjonction avec le fait que $\mathtt{DNF}(T) \mid t_{x_S}$ peut être dérivé en temps $O(|S| \cdot |T|)$, on en déduit que le nombre de modèles de $\mathtt{DNF}(T) \mid t_{x_S}$ peut être trouvé en $O(|S| \cdot |T|)$.

2.2 Explications abductives

Une explication abductive pour une instance $\boldsymbol{x}$ est un sous-ensemble de caractéristiques S tel que la restriction de l'instance $\boldsymbol{x}$ à S est suffisante pour obtenir la même prédiction que $\boldsymbol{x}$. Les arbres de décision sont localement explicables. Par construction, chaque instance $\boldsymbol{x}$ est

2. Un modèle est une interprétation des symboles d'une formule logique qui la rend vraie.

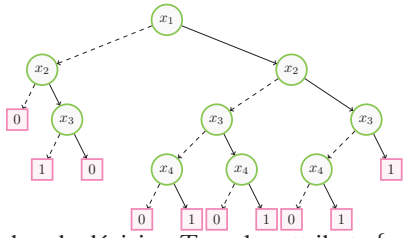

FIG. 1 – Un arbre de décision T sur les attributs $\{x_1, x_2, x_3, x_4\}$.

associée à un chemin unique de la racine à la feuille dans un arbre de décision. Une raison directe (Audemard et al., 2022) ou une explication de chemin (Izza et al., 2020) pour $\boldsymbol{x}$, étant donné l'arbre T, est un terme de la $\texttt{DNF}(T)$, noté $p_{\boldsymbol{x}}^T$, correspondant au chemin unique de la racine à la feuille de T. Cependant, ces raisons directes peuvent contenir un nombre arbitraire d'attributs redondants (Izza et al., 2020). Cela justifie la considération d'autres types d'explications abductives non redondantes pour les arbres de décision, telles que les raisons suffisantes (Darwiche et Hirth, 2020) et les raisons suffisantes de taille minimale. Formellement,

Définition 2 (Raison suffisante)**.** *Soit un classifieur $h : \{0,1\}^n \rightarrow \{0,1\}$ et $\boldsymbol{x} \in \{0,1\}^n$ tel que $h(\boldsymbol{x}) = 1$. Une raison suffisante pour $\boldsymbol{x}$ étant donné h est un impliquant premier t de h qui couvre $\boldsymbol{x}$. **Une raison suffisante de taille minimale** pour $\boldsymbol{x}$ étant donné h est une raison suffisante pour $\boldsymbol{x}$ étant donné h contenant un nombre minimal de littéraux.*

Les raisons suffisantes (Darwiche et Hirth, 2020) (ou PI-explications (Shih et al., 2018)) sont des explications abductives minimales par rapport à l'inclusion d'ensemble, et il a été démontré que, pour la classe $\texttt{DT}_n$, des raisons suffisantes peuvent être trouvées en temps polynomial (voir (Audemard et al., 2022)). Il est souvent pertinent de se concentrer sur les plus courtes (les raisons suffisantes de taille minimales), car la concision est généralement une propriété souhaitable des explications (principe du rasoir d'Occam). Cependant, trouver une raison suffisante de taille minimale s'avère difficile, en particulier pour les instances difficiles.

Proposition 1 (Audemard et al. (2022))**.** *Soient $T \in \texttt{DT}_n$ et $\boldsymbol{x} \in \{0,1\}^n$. Calculer une raison suffisante de taille minimale pour $\boldsymbol{x}$ étant donné T, est un problème* NP*-difficile.*

Malgré ce résultat, il est possible, dans de nombreux cas pratiques, de calculer une raison suffisante de taille minimale. Pour cela, les auteurs de (Audemard et al., 2022) ont exploité les progrès récents en optimisation combinatoire, notamment à travers l'utilisation des solveurs PARTIAL MAXSAT, qui permettent de calculer des raisons suffisantes de taille minimale. Cependant, lorsque la taille de l'arbre T et/ou la dimension de l'instance $\boldsymbol{x}$ est grande, le calcul d'une raison suffisante de taille minimale peut devenir hors de portée. Même avec des solveurs modernes, le temps de réponse peut être extrêmement long. C'est pourquoi nous nous tournons vers l'approximation des raisons suffisantes de taille minimale en utilisant la super-modularité.

Exemple 1. *Considérons un classifieur h représenté par l'arbre T de la figure 1 et une instance $\boldsymbol{x} = (1,1,1,1)$ telle que $T(\boldsymbol{x}) = 1$. Notons $\phi = \texttt{DNF}(T)$. Le nombre de modèles de ϕ (noté $w(\emptyset)$) est donné par $w(\emptyset) = 2^0 + 2^1 + 2^1 + 2^0 + 2^0 = 7$.*

Pour l'instance $\boldsymbol{x} = (1,1,1,1)$, nous observons que $h(\boldsymbol{x}) = 1$. La raison directe de $\boldsymbol{x}$ étant donné h est $p_{\boldsymbol{x}}^T = x_1 \wedge x_2 \wedge x_3$, $x_1 \wedge x_4$ et $x_1 \wedge x_2 \wedge x_3$ sont deux raisons suffisantes de $\boldsymbol{x}$. $x_1 \wedge x_4$ est l'unique raison suffisante de taille minimale pour $\boldsymbol{x}$. étant donné h.

3 Formulation du problème

La notion de raison suffisante de taille minimale t_{x_S} est souvent considérée comme un concept naturel pour expliquer le résultat d'un classifieur. Cependant, elle impose deux restrictions strictes : d'une part, elle exige que toutes les complétions d'une instance partielle $\boldsymbol{x}_S$ soient classées de la même manière, c'est-à-dire que la probabilité de commettre une « erreur d'explication » en utilisant l'instance partielle $\boldsymbol{x}_S$ au lieu de l'instance complète $\boldsymbol{x}$ soit égale à zéro. D'autre part, elle impose que la taille de t_{x_S} (ou simplement de S) soit minimale en termes de cardinalité, sachant que le nombre d'explications abductives pour une instance donnée, dans le cas d'un arbre de décision, peut être exponentiel (voir (Audemard et al., 2022)).

Nous définissons ensuite des notions qui nous seront utiles dans la suite de ce travail, notamment la fonction d'erreur d'explication $\epsilon_{h,\boldsymbol{x}}(S)$ étant donné un classifieur h et une instance $\boldsymbol{x}$, qui peut être interprétée comme la probabilité de commettre une "erreur d'explication" en utilisant un sous-ensemble S de caractéristiques. De plus, nous introduisons la notion de fonction super-modulaire, qui sera centrale dans notre étude d'approximation. Étant donné un classifieur h, et une instance $\boldsymbol{x}$ pour laquelle la prédiction $h(\boldsymbol{x})$ doit être expliquée, soit $\epsilon_{h,\boldsymbol{x}} : 2^{[n]} \rightarrow \mathbb{R}$ la *fonction d'erreur d'explication* (Bounia et Koriche, 2023) est définie par

$$\epsilon_{h,\boldsymbol{x}}(S) = \frac{|\{\boldsymbol{y} \in \{0,1\}^n : h(\boldsymbol{y}) \neq h(\boldsymbol{x}), \boldsymbol{y}_S = \boldsymbol{x}_S\}|}{|\{\boldsymbol{y} \in \{0,1\}^n : \boldsymbol{y}_S = \boldsymbol{x}_S\}|} = \frac{\mu_{h,\boldsymbol{x}}(S)}{2^{n-|S|}} \tag{1}$$

Comme indiqué ci-dessus, $\epsilon_{h,\boldsymbol{x}}(S)$ peut être interprétée comme la probabilité de commettre une « erreur d'explication » où $\mu_{h,\boldsymbol{x}}(S)$ est interprété comme le nombre d'erreurs induites par le choix de S. Pour un sous-ensemble de caractéristiques S, t_{x_S} est une explication abductive pour $\boldsymbol{x}$, étant donné h, si $\epsilon_{h,\boldsymbol{x}}(S) = 0$. De plus, t_{x_S} est une raison suffisante si $\epsilon_{h,\boldsymbol{x}}(S) = 0$ et $\epsilon_{h,\boldsymbol{x}}(S') > 0$ pour tout sous-ensemble propre S' de S. Notons que, lorsque h est représenté par un arbre de décision T, alors $\mu_{h,\boldsymbol{x}}(S)$ dans (1) peut être réécrit simplement comme suit :

$$\mu_{h,\boldsymbol{x}}(S) = \begin{cases} \sum\limits_{t \in \texttt{DNF}(T)|t_{x_S}} 2^{n-|t|} & \text{si } h(\boldsymbol{x}) = 0 \\ 2^{n-|S|} - \sum\limits_{t \in \texttt{DNF}(T)|t_{x_S}} 2^{n-|t|} & \text{si } h(\boldsymbol{x}) = 1 \end{cases}$$

Le résultat montre que l'évaluation de $\epsilon_{h,x}(S)$ peut être effectuée en un temps $O(|S| \cdot |T|)$ lorsque h est un arbre de décision T, ce qui n'est pas toujours le cas générale. En effet, dans le cas général, le problème d'évaluer $\epsilon_{h,x}(S)$ est $\#$**-difficile** (Wäldchen et al., 2021).

Nous formulons maintenant le problème de la recherche d'une raison suffisante de taille minimale. L'idée derrière repose sur le fait qu'un terme t_{x_S} associé à un sous-ensemble de caractéristiques $S \subseteq [d]$ est une explication abductive de $\boldsymbol{x}$ étant donné h si $\epsilon_{h,\boldsymbol{x}}(S) = 0$, ce qui est équivalent à $\mu_{h,\boldsymbol{x}}(S) = 0$. Ainsi, par définition, une explication abductive de taille minimale est simplement le plus petit ensemble en cardinalité qui vérifie $\mu_{h,\boldsymbol{x}}(S) = 0$.

Approximation d'une raison suffisante de taille minimale. Le problème de la recherche d'une raison suffisante de taille minimale pour une instance $\boldsymbol{x}$, étant donné un classifieur h, peut être formulé comme un problème de sélection de leaders (ou k-leaders). Plus précisément, notre objectif est de sélectionner un sous-ensemble S de leaders de taille minimale (contrainte

hard) qui satisfait une certaine contrainte sur la fonction d'erreur d'explication $\epsilon_{h,x}(.) \leq 0$ (contrainte soft), ce qui équivaut à écrire $\mu_{h,x}(.) \leq 0$.

Problème 1. *Soit un classifieur h et une instance $\boldsymbol{x}$, le problème de calculer une raison suffisante de taille minimale pour h et $\boldsymbol{x}$ peut être formulé comme suit :*

$$\min_{S \subseteq [n]} |S| \qquad (\mathbb{P}_1) \qquad \textbf{\textit{s.t.}} \quad \mu_{h,x}(S) \leq 0$$

Proposition 2. *Soit un classifieur $h : \{0,1\}^n \rightarrow \{0,1\}$ et une instance $\boldsymbol{x} \in \{0,1\}^n$, ainsi qu'un sous-ensemble $S^* \subseteq [n]$ de caractéristiques. S^* est une solution optimale du problème 1 si et seulement si $t_{x_{S^*}}$ est une raison suffisante de taille minimale pour $\boldsymbol{x}$ étant donné h.*

En général, la sélection d'un ensemble de leaders de taille minimale pour une fonction super-modulaire, comme dans le problème 1, est NP-difficile (Clark et al., 2014b). Cela concorde avec la proposition 2 et le fait que le calcul d'une raison suffisante de taille minimale est également NP-difficile lorsque h est représenté par un arbre de décision. Toutefois, une approximation efficace de cette raison peut être obtenue en exploitant l'optimisation super-modulaire. La fonction $\mu_{h,x}(.)$ étant super-modulaire décroissante (Bounia et Koriche, 2023), permet d'élaborer un algorithme qui approche l'ensemble optimal de leaders, avec une garantie sur l'écart d'optimalité entre la taille de l'ensemble sélectionné $|S|$ et la solution optimale S^*. Cet algorithme satisfait également la contrainte $\epsilon_{h,x}(S) = 0$, équivalente à $\mu_{h,x}(S) = 0$.

4 Algorithmes d'approximation

L'idée principale de cette étude est de relâcher l'exigence de trouver une solution optimale au problème 1 et de se contenter d'une solution « suffisamment bonne », en utilisant la minimisation super-modulaire. Nous introduisons d'abord certains notions de base.

Fonctions super-modulaires. Soit une fonction d'ensemble à valeurs réelles $f : 2^{[n]} \rightarrow \mathbb{R}$. f est dite croissante si $f(S \cup \{i\}) \geq f(S)$ pour tout $S \subseteq [n]$ et $i \in [n] \setminus S$, et décroissante si $f(S \cup \{i\}) \leq f(S)$ pour tout $S \subseteq [n]$ et $i \in [n] \setminus S$. f est dite super-modulaire si elle satisfait pour tout sous-ensembles A, B de $[n]$ la condition suivante : $f(A \cup B) + f(A \cap B) \geq f(A) + f(B)$. D'une manière duale, f est sous-modulaire si, pour tous les sous-ensembles A et B de $[n]$: $f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$. Pour tout $S \subseteq [n]$ et $i \in S$, f est super-modulaire si et seulement si $-f$ est sous-modulaire. Enfin, f est modulaire si elle est à la fois sous-modulaire et super-modulaire.

En général, la fonction d'erreur $\epsilon_{h,x}(\cdot)$ n'est ni super-modulaire, ni sous-modulaire, et elle n'est ni croissante, ni décroissante (Bounia et Koriche, 2023). Cependant, si l'on se concentre sur la version non-normalisée $\mu_{h,x}(\cdot)$, les propriétés suivantes peuvent être dérivées.

Proposition 3. *Soit $h : \{0,1\}^n \rightarrow \{0,1\}$ un classifieur, $x \in \{0,1\}^n$ une instance. Alors, $\mu_{h,x}$ est une fonction positive, super-modulaire et décroissante.*

La proposition 3 implique que le problème 1 est un problème d'optimisation super-modulaire. Bien que les problèmes d'optimisation super-modulaire de cette forme soient généralement NP-difficiles (Clark et al., 2014b), un algorithme glouton retournera un ensemble S dont la taille $|S|$ est proche de celle de la solution optimale $|S^*|$.

4.1 Algorithme glouton.

La super-modularité de $\mu_{h,x}(.)$ permet de calculer efficacement une solution approximative au problème 1 à l'aide d'un algorithme glouton. L'algorithme commence par initialiser l'ensemble des caractéristiques leaders à $S_0 = \emptyset$. À chaque itération, l'élément e_i^* qui maximise $\mu_{h,x}(S) - \mu_{h,x}(S_{i-1} \cup \{e^i\})$ est ajouté, de sorte que $S_i = S_{i-1} \cup \{e_i^*\}$. L'algorithme se termine lorsque $\mu_{h,x}(S_k) \leq 0$ et renvoie l'ensemble $S = S_i$. Une description en pseudo-code de cet algorithme est donnée sous le nom `static-`α (Clark et al., 2014a).

Algorithm 1

Input: un classifieur h représenté par un arbre T, une instance $\boldsymbol{x}$, erreur de terminaison α
Output: un ensemble d'attributs leaders S (un impliquant de $\boldsymbol{x}$ étant donné T)
Initialisation : $S \leftarrow \emptyset$, error $\leftarrow \alpha$ ($\alpha > 0$), $V = [n]$
while *error* > 0 **do**
  $e^* \longleftarrow \underset{e \in V \setminus S}{\operatorname{argmax}} \mu_{h,x}(S) - \mu_{h,x}(S \cup \{e\})$
  if $\mu_{h,x}(S) - \mu_{h,x}(S \cup \{e\}) \leq 0$ **then**
    return S; **exit**
  else
    $S \longleftarrow S \cup \{e^*\}$
    error $\longleftarrow \mu_{h,x}(S)$
return S; **exit**

Le théorème 1 donne les bornes d'approximation pour la solution renvoyée par l'algorithme 1.

Théorème 1. *Soit $h : \{0,1\}^n \rightarrow \{0,1\}$ un classifieur et $\boldsymbol{x} \in \{0,1\}^n$ une instance. Notons $|S^*| = k^*$ la solution optimale du problème 1, et soit $|S| = k$ l'ensemble renvoyé par l'algorithme 1. Alors, $\frac{|S|}{|S^*|} = \frac{k}{k^*} \leq 1 + \ln\left(\frac{\mu_{max}}{\mu_{h,x}(S_{k-1})}\right)$. Où $\mu_{max} = \max_{i \in [n]} \ \mu_{h,x}(\{i\})$.*

Le théorème 1 établit une borne d'approximation en taille, garantissant que la solution S trouvée par l'algorithme 1 est proche de la solution optimale. Cependant, bien que cette solution soit un impliquant de $\boldsymbol{x}$ étant donné h, elle peut inclure des attributs non pertinents, ce qui signifie qu'elle n'est pas nécessairement une raison suffisante (explication minimale pour l'inclusion). Néanmoins, par construction, la valeur de $\mu_{h,x}(S_{k-1})$ ne peut pas être nulle.

Proposition 4. *Soit un classifieur h représenté par un arbre de décision $T \in \mathtt{DT}_n$ et une instance $\boldsymbol{x} \in \{0,1\}^n$. L'algorithme 1 s'exécute en temps $O(n^3 \cdot |T|)$.*

Exemple 2. *Pour le classifieur h et l'instance $\boldsymbol{x}$ de l'exemple 1. Nous cherchons une approximation d'une raison suffisante de taille minimale. Les étapes de l'algorithme 1 sont : $x_1 = \underset{e \in \{x_1,x_2,x_3,x_4\}}{\operatorname{argmax}} \mu_{h,x}(\emptyset) - \mu_{h,x}(\{e\})$, alors $S = \{x_1\}$ et $x_4 = \underset{e \in \{x_2,x_3,x_4\}}{\operatorname{argmax}} \mu_{h,x}(S) - \mu_{h,x}(S \cup \{e\})$. On obtient alors $S = \{x_1, x_4\}$ et $\mu_{h,x}(\{x_1, x_4\}) = 0$. Ainsi, l'algorithme 1 a trouvé une raison suffisante de taille minimale (la solution optimale du problème 1).*

Amélioration de la sortie de l'algorithme 1. Nous savons que la sortie de l'algorithme 1 ne constitue pas nécessairement une raison suffisante. Afin d'améliorer la sortie de notre algorithme, nous pouvons extraire une raison suffisante à partir de la solution S renvoyée par l'algorithme 1 en utilisant un simple algorithme glouton décrit par l'algorithme 2.

Algorithm 2 : Dérivation d'une raison suffisante

Input: un classifieur h représenté par un arbre T, $\boldsymbol{x} \in \{0,1\}^n$, un ensemble S
Output: une raison suffisante pour $\boldsymbol{x}$ étant donné T
$I \leftarrow S$ /* S est la sortie de l'algorithme 1 */
for $l \in I$ **do**
  if $\mu_{h,x}(S) = 0$ **then**
    $S \leftarrow S - \{l\}$
return S

5 Expérimentations

Afin de valider l'efficacité de nos algorithmes, nous avons considéré diverses instances du problème 1, où le classifieur en entrée est décrit par un arbre de décision. Le code a été écrit en langage Python. Toutes les expériences ont été réalisées sur un ordinateur équipé d'un processeur Intel(R) Core $i9-9900$ à 3.1 GHz et de 64 GiB de RAM.

dataset			decision tree			|Explanation|		
name	#F	#I	%A	$\|T\|$	Depth	S^*	S_{algo1}	S_{improve}
placement	10	215	92.31	11	6	3.71 ($\pm$ 0.99)	3.71 ($\pm$ 0.99)	3.71 ($\pm$ 0.99)
letter	91	20000	99.08	144	15	6.76 ($\pm$ 1.33)	6.82 ($\pm$ 1.41)	6.81 ($\pm$ 1.4)
diabetes t	101	768	68.83	107	15	7.85 ($\pm$ 2.36)	7.85 ($\pm$ 2.36)	7.85 ($\pm$ 2.36)
student.por	27	649	89.23	30	8	4.22 ($\pm$ 0.92)	4.22 ($\pm$ 0.92)	4.22 ($\pm$ 0.92)
ad-data	115	3279	96.44	135	83	28.31 ($\pm$ 9.77)	29.06 ($\pm$ 10.13)	29.03 ($\pm$ 10.12)
balance	16	625	85.64	88	11	4.89 ($\pm$ 0.84)	5.04 ($\pm$ 0.95)	5.04 ($\pm$ 0.95)
breast. c	31	286	62.79	80	16	7.15 ($\pm$ 2.58)	7.51 ($\pm$ 3.03)	7.17 ($\pm$ 2.58)
christine	327	5418	62.12	328	38	15.77 ($\pm$ 9.87)	15.77 ($\pm$ 9.87)	15.77 ($\pm$ 9.87)
haberman	55	306	69.57	75	13	6.57 ($\pm$ 1.33)	6.59 ($\pm$ 1.33)	6.59 ($\pm$ 1.33)
spambase	219	4601	90.51	222	32	16.46 ($\pm$ 6.87)	16.46 ($\pm$ 6.87)	16.46 ($\pm$ 6.87)
meta	40	528	90.57	66	15	4.13 ($\pm$ 1.42)	4.14 ($\pm$ 1.43)	4.14 ($\pm$ 1.43)
bank	805	41188	88.93	2090	25	14.06 ($\pm$ 2.21)	14.1 ($\pm$ 2.23)	14.08 ($\pm$ 2.22)

TAB. 1 – Statistiques sur la fiabilité des solutions du problème 1 générées par l'algorithme 1.

Nous avons mené plusieurs expériences pour évaluer les performances de l'algorithme 1, en nous concentrant sur deux objectifs principaux : mesurer l'exactitude de l'approximation de la raison suffisante de taille minimale en comparant les résultats obtenus par une méthode exacte avec ceux fournis par l'algorithme 1 et l'algorithme 2, et démontrer l'efficacité de notre approche sur des instances difficiles en comparant les temps de calcul nécessaires entre les méthodes SAT et l'algorithme 1, les résultats montrant clairement que ce dernier constitue une alternative très efficace lorsque les méthodes SAT deviennent inefficaces.

5.1 Protocole Expérimental

Nous avons étudié un ensemble de 18 jeux de données, provenant de sources reconnues telles que **Kaggle, OpenML** et **UCI**. Les attributs catégoriels ont été traités comme des entiers, et les attributs numériques ont été binarisés lors de l'apprentissage des arbres de décision. Dans le but de présenter un cas avec de nombreuses instances difficiles pour démontrer l'efficacité de nos algorithmes, un pré-traitement spécifique pour *sentiment140* a été réalisé. Le protocole est détaillé dans le matériel supplémentaire, en raison de contraintes d'espace.

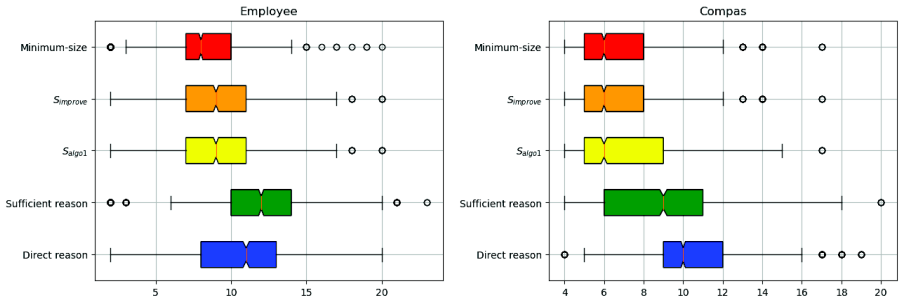

FIG. 2 – Boîtes à moustaches des tailles pour « employee » (à gauche) et « compas » (à droite).

Pour chaque instance $\boldsymbol{x}$ de l'ensemble de test d'un dataset b, T_b est l'arbre de décision appris à partir de l'ensemble d'entraînement en utilisant l'algorithme CART via l'implémentation de *Scikit-Learn*. Sa précision est évaluée sur cet ensemble de test. Nous avons comparé les tailles moyennes des solutions approximatives des $m = 1000$ instances tirées aléatoirement de cet ensemble avec celles des différents types d'explications (voir Figure 1 et tableau 2).

Pour évaluer les performances de nos algorithmes gloutons, nous avons comparé les solutions retournées par l'algorithme 1 et l'algorithme 2 à celles d'une méthode exacte de résolution du problème 1. Nous avons utilisé la méthode décrite dans (Audemard et al., 2022), qui s'appuie sur le solveur PARTIAL MAXSAT **RC2**, via son implémentation dans la bibliothèque **Pyxai** (Audemard et al., 2023), pour obtenir une raison suffisante de taille minimale pour chaque instance $\boldsymbol{x}$ étant donné T_b. Pour évaluer l'efficacité de nos solutions approximatives, nous avons comparé les tailles moyennes de la solution optimale S^* à celles d'une raison suffisante S_r, de la raison directe (Audemard et al., 2022) (ou explication du chemin (Izza et al., 2020)) P_x^T, ainsi que des solutions fournies par l'algorithme 1 (S_{alg1}) et 2 sa version améliorée S_{improve}, et comparé leurs tailles respectives (voir tableau 1 et figure 2).

Ensuite, nous nous concentrons sur les instances où le calcul de la solution optimale du problème 1 devient difficile, même avec des encodages basés sur un solveur SAT (Arenas et al., 2022) ou PARTIAL MAXSAT (Audemard et al., 2022). Ces instances, caractérisées par leur structure complexe ou leur grande taille, sont particulièrement difficiles à résoudre. Pour cela, nous avons entraîné un arbre de décision sur des benchmarks b de grande dimension (n élevé) et de profondeur importante (**Depth**). Nous avons ensuite comptabilisé le nombre d'instances parmi les $m = 10^3$ sélectionnées dans l'ensemble de test pour lesquelles les méthodes exactes ne trouvent pas de solution dans un temps limité, fixé respectivement à 1, 3, et 5 minutes. Ces résultats ont été comparés aux temps de calcul moyens nécessaires à l'algorithme 1.

5.2 Résultats expérimentaux

Le tableau 1 présente un extrait de nos résultats pour 15 ensembles de données. Pour chacun, le temps moyen pour obtenir une raison suffisante de taille minimale (notée $|S^*|$) ne dépasse pas 10 secondes. La première colonne indique le nom de l'ensemble de données b, suivi par le nombre d'attributs binaires ($\#F$), le nombre d'instances ($\#I$), la précision de l'arbre T_b ($\%A$), la taille de l'arbre ($|T|$), et sa profondeur (Depth). La colonne |Explanation| montre la taille moyenne des explications calculées sur les m instances : $|S^*|$, $|S_{\text{algo1}}|$, et $|S_{\text{improve}}|$, cor-

respondant respectivement à la taille moyenne des solutions approximatives de l'algorithme 1 et après amélioration par l'algorithme 2. Nous constatons que la taille moyenne des solutions de 1 est très proche de celle des raisons suffisantes minimales. En général, pour tous les jeux de données, l'erreur moyenne $||S_{\text{algo1}}| - |S^*||$ est de l'ordre de 10^{-2}. Cette erreur diminue légèrement avec S_{improve}, atteignant 10^{-3}. Ces résultats montrent que l'algorithme 1 offre déjà d'excellentes performances pour l'approximation des raisons suffisantes minimales du problème 1.

Dataset	#I	#F	%A	$\|T\|$	Depth	1-min	3-min	5-min
sentiment140	$\approx 10^6$	47877	83.28	14537	93	$(41, -)$	$(35, -)$	$(33, -)$
adult	48842	2395	81.68	5014	54	$(11, 712)$	$(7, 623)$	$(3, 557)$
gisette	13500	5000	93.14	436	32	$(3, 11)$	$(0, 3)$	$(0, 0)$

TAB. 2 – Tableau des résultats pour les instances difficiles de 3 ensembles de données.

Le tableau 2 présente les résultats obtenus pour trois ensembles de données de grande dimension, utilisés pour entraîner un modèle visant à générer des arbres de grande taille avec une profondeur maximale. Les colonnes $\#I$, $\#F$, et $\%A$, $|T|$, Depth indiquent respectivement le nombre d'instances, d'attributs binaires et la précision de l'arbre, la taille et la profondeur de l'arbre. Les colonnes 1-min, 3-min, et 5-min montrent le nombre d'instances pour lesquelles les solveurs (PARTIAL MAXSAT, SAT) n'ont pas retourné de résultats dans des temps limites respectifs de 1 min, 3 min, et 5 min. Notons que l'algorithme 1 fournit une réponse en moins de 30 secondes pour tous les ensembles de données et instances. Bien que le solveur PARTIAL MAXSAT réussisse à donner des réponses pour de nombreuses instances en moins d'une minute, certaines instances considérées comme difficiles ne reçoivent pas de réponse en 5 min. Par exemple et sur 1000 instances, pour *adult*, seules 3 ne donnent pas de résultats en moins de 5 minutes (contre 557 pour le solveur SAT), tandis que pour *sentiment140*, le solveur SAT ne passe pas à l'échelle. En effet, l'encodage SAT rencontre des difficultés pour de nombreuses d'instances, en raison de l'explosion du nombre de clauses de l'encodage. Il est mentionné dans ((Bounia et Koriche, 2023; Arenas et al., 2022)) que ce phénomène se produit lorsque l'arbre est trop complexe ou lorsque l'entrée est de grande dimension. Dans ces cas, l'approximation via l'algorithme 1 s'avère particulièrement utile.

Pour illustrer le gain et la perte d'intelligibilité des approximations générées par les algorithmes 1 et 2, nous avons créé des boxplots pour les jeux de données « compas » et « spambase », montrant la distribution des tailles de différentes explications abductives : raisons directes en « bleu », raisons suffisantes en « vert », sorties de l'algorithme 1 en « jaune », de l'algorithme 2 en « orange », et de la raison suffisante de taille minimale en « rouge ». La figure 2 présente ces boxplots. Nous constatons une réduction significative du nombre d'attributs utilisés dans les raisons directes et suffisantes grâce aux solutions approximatives de 1, surtout dans sa version améliorée, qui constitue une raison suffisante. Cela confirme que les raisons suffisantes minimales approximatives peuvent être suffisamment concises pour constituer une alternative efficace dans les cas où le calcul d'une raison suffisante minimale est hors de portée.

6 Conclusion

Dans cet article, nous avons abordé le problème de l'approximation des explications abductives de taille minimale, formulé comme la sélection d'un ensemble minimal de leaders atteignant une borne d'erreur pour une fonction super-modulaire (problème 1). Trouver une explication abductive de taille minimale pour une instance $\boldsymbol{x}$ et un classifieur h peut s'avérer difficile, même lorsque h est représenté par un arbre de décision T, surtout pour les instances complexes où les méthodes exactes sont inefficaces. Pour répondre à cette difficulté, nous avons proposé un algorithme glouton (algorithme 1) avec une borne d'approximation, permettant de calculer efficacement des explications abductives de taille quasi-minimale. Nos expérimentations ont montré que cet algorithme, combiné à l'algorithme 2, génère des explications proches de la taille minimale, tout en étant plus compactes qu'une raison suffisante arbitraire ou une raison directe pour une instance $\boldsymbol{x}$ donnée T. En équilibrant temps de calcul et taille de l'explication, ces algorithmes s'avèrent efficaces pour les instances difficiles. Pour nos travaux futurs, nous envisageons d'étendre cette approche à d'autres types de classifieurs, notamment les forêts aléatoires, les arbres boostés et les réseaux de neurones (y compris les MLP), où l'évaluation de la fonction d'erreur non-normalisée $\mu_{h,x}$ est coûteuse.

Références

Arenas, M., P. Barceló, M. Romero Orth, et B. Subercaseaux (2022). On computing probabilistic explanations for decision trees. *Advances in Neural Information Processing Systems 35*.

Audemard, G., S. Bellart, L. Bounia, F. Koriche, J. Lagniez, et P. Marquis (2021). On the computational intelligibility of boolean classifiers. In *Proc. of KR'21*, pp. 74–86.

Audemard, G., S. Bellart, L. Bounia, F. Koriche, J.-M. Lagniez, et P. Marquis (2022). On the explanatory power of boolean decision trees. *Data & Knowledge Engineering 142*, 102088.

Audemard, G., S. Bellart, L. Bounia, J.-M. Lagniez, P. Marquis, et N. Szczepanski (2023). Pyxai : calculer des explications pour des modèles d'apprentissage supervisé. *EGC*.

Bounia, L. et F. Koriche (2023). Approximating probabilistic explanations via supermodular minimization. In *Uncertainty in Artificial Intelligence (UAI 2023)*, Volume 216.

Breiman, L. (2001). Random forests. *Machine Learning 45*(1), 5–32.

Chen, T. et C. Guestrin (2016). Xgboost : A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, pp. 785–794.

Clark, A., B. Alomair, L. Bushnell, et R. Poovendran (2014a). Minimizing convergence error in multi-agent systems via leader selection : A supermodular optimization approach. *IEEE Transactions on Automatic Control 59*(6), 1480–1494.

Clark, A., L. Bushnell, et R. Poovendran (2014b). A supermodular optimization framework for leader selection under link noise in linear multi-agent systems. *IEEE Transactions on Automatic Control 59*(2), 283–296.

Crama, Y. et P. L. Hammer (2011). Boolean functions - theory, algorithms, and applications. In *Encyclopedia of mathematics and its applications*.

Darwiche, A. (1999). Compiling devices into decomposable negation normal form.

Darwiche, A. et A. Hirth (2020). On the reasons behind decisions. In *Proc. of ECAI'20*.

Frosst, N. et G. E. Hinton (2017). Distilling a neural network into a soft decision tree. *ArXiv*.

Ignatiev, A., N. Narodytska, et J. Marques-Silva (2019). Abduction-based explanations for machine learning models. In *Proc. of AAAI'19*, pp. 1511–1519.

Izza, Y., A. Ignatiev, et J. Marques-Silva (2020). On explaining decision trees. *ArXiv*.

Izza, Y. et J. Marques-Silva (2021). On explaining random forests with SAT. In *Proc. of IJCAI'21*, pp. 2584–2591.

Lipton, Z. C. (2018). The mythos of model interpretability. *CACM 61*(10), 36–43.

Louenas, B. (2023). *Modéles formels pour l'IA explicable : des explications pour les arbres de décision*. Ph. D. thesis, Université d'Artois.

Lundberg, S. et S.-I. Lee (2017). A unified approach to interpreting model predictions. In *Proc. of NIPS'17*, pp. 4765–4774.

Miller, T. (2019). Explanation in artificial intelligence : Insights from the social sciences. *Artificial Intelligence 267*, 1–38.

Molnar, C. (2019). *Interpretable Machine Learning - A Guide for Making Black Box Models Explainable*. Leanpub.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning 1*(1), 81–106.

Ribeiro, M. T., S. Singh, et C. Guestrin (2016). "why should I trust you ?" : Explaining the predictions of any classifier. In *Proc. of SIGKDD'16*, pp. 1135–1144.

Ribeiro, M. T., S. Singh, et C. Guestrin (2018). Anchors : High-precision model-agnostic explanations. In *Proc. of AAAI'18*, pp. 1527–1535.

Shih, A., A. Choi, et A. Darwiche (2018). A symbolic approach to explaining bayesian network classifiers. In *Proc. of IJCAI'18*, pp. 5103–5111.

Wäldchen, S., J. Macdonald, S. Hauch, et G. Kutyniok (2021). The computational complexity of understanding binary classifier decisions. *J. Artif. Intell. Res. 70*, 351–387.

Summary

In this work, we address the problem of approximating minimum-size abductive explanations for decision trees using supermodular optimization. Finding a minimum-size abductive explanation is a time-intensive task, even for restricted classifiers like decision trees. This problem, being NP-**hard** for decision trees, makes exact approaches particularly time-consuming, especially for complex instances and high-dimensional data. To overcome these challenges, approximate or heuristic methods are often preferred, allowing for reduced computational load and faster results, albeit sometimes at the cost of optimality. Here, we propose a greedy algorithm to efficiently approximate minimum-size abductive explanations. Our experiments show that for difficult-to-explain instances, this algorithm provides an effective alternative to exact methods based on SAT encodings.