Explicabilité en Intelligence Artificielle

Notes de cours

Nadjib Lazaar
Université Paris-Saclay
lazaar@lisn.fr

Introduction

L'explicabilité en intelligence artificielle est un domaine crucial qui vise à rendre les décisions des modèles d'IA compréhensibles par les humains. Avec l'essor des systèmes d'apprentissage profond et des modèles complexes, comprendre pourquoi une décision est prise devient fondamental pour garantir la confiance, la transparence et l'équité des systèmes automatisés.
Dans ce cours, nous allons explorer différentes approches de l'explicabilité et introduire des concepts fondamentaux comme les ensembles minimaux d'explication. Nous nous concentrerons ensuite sur l'algorithme QuickXplain, une méthode efficace permettant d'extraire un sous-ensemble minimal de contraintes expliquant une contradiction dans un problème donné.

1 Introduction

L'intelligence artificielle (IA) occupe aujourd'hui une place centrale dans de nombreux domaines tels que la santé, la finance, le transport ou encore la sécurité. Si les performances des systèmes basés sur l'IA, notamment ceux utilisant l'apprentissage automatique, sont impressionnantes, leur adoption à large échelle soulève une question essentielle : **peut-on leur faire confiance ?**

C'est dans ce contexte qu'émerge le besoin croissant d'**explicabilité**. Cette notion recouvre l'ensemble des méthodes, techniques et approches qui permettent de comprendre, interpréter et justifier les décisions prises par des systèmes d'IA. L'explicabilité est devenue un enjeu à la fois technique, éthique et juridique.

Les motivations pour rendre l'IA explicable sont multiples :

— **Confiance** : Un utilisateur est plus enclin à accepter une décision s'il peut en comprendre la justification.

— **Vérification et validation** : Les experts doivent pouvoir analyser les comportements des modèles, notamment en cas de dysfonctionnement.
— **Responsabilité** : Dans des contextes critiques (justice, médecine, transport autonome), il est nécessaire de pouvoir identifier les causes d'une décision.
— **Conformité réglementaire** : Des textes comme le RGPD ou l'AI Act européen exigent des garanties en matière de transparence et de contrôle.

Cette note de cours a pour objectif de présenter les principales approches de l'explicabilité en IA, depuis les méthodes générales jusqu'aux fondements logiques basés sur la cohérence des systèmes de contraintes. Nous aborderons notamment :

— les motivations et le cadre légal (RGPD, AI Act) ;
— les grandes familles de méthodes d'explicabilité ;
— les approches fondées sur la logique et la programmation par contraintes : incohérence, MUS/MCS/MSS, et l'algorithme QuickXplain.

2 Cadre réglementaire : RGPD et AI Act

L'explicabilité en IA ne se limite pas à un besoin technique : elle est aussi une exigence légale. Deux textes majeurs encadrent aujourd'hui les usages de l'IA en Europe : le **Règlement Général sur la Protection des Données (RGPD)** et le **AI Act**, encore en cours d'adoption formelle mais déjà très structurant.

2.1 Le RGPD et le droit à l'explication

Le RGPD, entré en vigueur en 2018, encadre le traitement automatisé des données personnelles. L'article 22 introduit une limite importante : un individu ne peut faire l'objet d'une décision entièrement automatisée produisant des effets juridiques à son égard, sauf conditions strictes.

L'un des apports les plus discutés du RGPD est l'idée de **droit à l'explication**, bien qu'il ne soit pas explicitement formulé ainsi. L'article 15(1)(h) prévoit que la personne concernée puisse obtenir « des informations utiles concernant la logique sous-jacente » d'un traitement automatisé.

Implications pour l'IA :
— Obligation de transparence sur les finalités et la logique utilisée.
— Nécessité d'offrir des garanties contre les biais et les discriminations.
— Importance des méthodes interprétables ou explicables.

2.2 L'AI Act : vers une régulation spécifique de l'IA

Le **AI Act** est un règlement européen en cours de finalisation (version consolidée fin 2023), visant à encadrer le développement et le déploiement des systèmes d'IA sur le territoire européen. Il introduit une classification des systèmes d'IA selon leur niveau de risque : *minimal, limité, élevé, inacceptable.*

Pour les systèmes à haut risque, des exigences strictes sont imposées, notamment :

1. **Documentation technique** claire et à jour ;
2. **Données d'apprentissage de qualité** ;
3. **Traçabilité des décisions** ;
4. **Explicabilité et transparence** des résultats ;
5. **Surveillance humaine** ;
6. **Robustesse, précision et cybersécurité** ;
7. **Gestion des risques** tout au long du cycle de vie.

Ces exigences rendent incontournable le recours à des approches d'explicabilité, que ce soit au moment de la conception (explicabilité intrinsèque) ou de l'usage (explicabilité post-hoc).

2.3 Vers une explicabilité normative

Les textes comme le RGPD et l'AI Act ne définissent pas exactement ce qu'est une bonne explication, mais ils posent des exigences de *transparence*, *compréhensibilité*, *justification* et *redevabilité*. Il revient donc à la recherche de proposer des méthodes adaptées, en fonction du contexte d'utilisation, du public cible, et des risques associés.

3 Méthodes d'explicabilité en intelligence artificielle

Les approches d'explicabilité visent à rendre les modèles d'intelligence artificielle compréhensibles pour les humains. On distingue deux grandes dimensions : **le moment** où intervient l'explication (avant ou après l'apprentissage) et **le niveau** de l'explication (globale ou locale).

3.1 Typologie des méthodes

Explicabilité intrinsèque Certains modèles sont, par nature, interprétables. C'est le cas des arbres de décision, des règles logiques, ou des modèles linéaires. L'explication est alors directement liée à la structure du modèle.

Explicabilité post-hoc Lorsqu'un modèle est complexe (réseaux de neurones profonds, forêts aléatoires, etc.), des méthodes d'explication sont nécessaires après l'apprentissage. On cherche alors à expliquer une décision particulière sans modifier le modèle.

Explication locale vs globale

- **Locale** : expliquer une prédiction particulière (ex : pourquoi ce patient a-t-il reçu ce diagnostic ?).
- **Globale** : comprendre le fonctionnement général du modèle (ex : quelles sont les variables les plus importantes ?).

3.2 Quelques méthodes classiques

- **LIME** (Local Interpretable Model-agnostic Explanations) : crée un modèle simple local autour de la prédiction à expliquer.
- **SHAP** (SHapley Additive exPlanations) : s'appuie sur la théorie des jeux pour attribuer une importance à chaque caractéristique.
- **Contre-exemples** : indique ce qu'il faudrait changer pour que la prédiction soit différente.
- **Saliency maps** (en vision par ordinateur) : mettent en évidence les zones d'une image ayant influencé la décision.

3.3 Limites des approches classiques

Ces méthodes sont utiles, mais elles présentent des limites :

- Manque de rigueur formelle ;
- Difficulté à vérifier la validité ou la complétude des explications ;
- Parfois peu interprétables pour un non-expert ;
- Sensibilité aux perturbations (instabilité).

3.4 Vers une explicabilité fondée sur la logique

Face à ces limites, une autre approche se développe : **l'explicabilité logique ou symbolique**. Elle repose sur des systèmes formels (logique propositionnelle, programmation par contraintes, ontologies) pour produire des explications vérifiables, auditables et souvent plus compréhensibles.

Ces méthodes permettent notamment d'expliquer l'**incohérence** dans un système (ex : des contraintes qui ne peuvent pas être satisfaites simultanément), en identifiant les parties minimales responsables du problème.

C'est dans ce cadre que s'inscrivent les notions de **MUS (Minimal Unsatisfiable Subsets)**, **MCS (Minimal Correction Subsets)**, et **MSS (Maximal Satisfiable Subsets)**, ainsi que l'algorithme **QuickXplain**, que nous allons explorer dans les prochaines sections.

4 Explication par incohérence : MUS, MCS et MSS

Dans le cadre de l'explicabilité en IA, notamment pour les systèmes symboliques ou à base de contraintes, une approche puissante consiste à expliquer une

incohérence observée dans un système. Une incohérence survient lorsqu'un ensemble de contraintes ou de connaissances est **insatisfiable**, c'est-à-dire qu'il n'existe aucune solution qui satisfait l'ensemble.

Pour comprendre *pourquoi* une telle incohérence existe, on peut chercher les sous-ensembles de contraintes responsables. Cela nous conduit aux notions clés de :

— **MUS** : *Minimal Unsatisfiable Subsets*
— **MCS** : *Minimal Correction Subsets*
— **MSS** : *Maximal Satisfiable Subsets*

Ces objets permettent de produire des **explications minimales** de l'incohérence.

4.1 Notations de base

— Soit C un ensemble fini de contraintes ou formules.
— On dit que C est **insatisfiable** s'il n'existe aucune affectation qui satisfait simultanément toutes les contraintes de C.
— On note $\mathsf{SAT}(S)$ pour dire que l'ensemble $S \subseteq C$ est satisfiable.

4.2 Définition d'un MUS

Définition 1 (MUS)**.** *Soit C un ensemble de contraintes tel que C est insatisfiable. Un sous-ensemble $M \subseteq C$ est un **MUS** (Minimal Unsatisfiable Subset) si :*

— *M est insatisfiable : $\mathsf{SAT}(M) = \mathsf{false}$*
— *$\forall c \in M,\ \mathsf{SAT}(M \setminus \{c\}) = \mathsf{true}$ (minimalité)*

Intuition : Un MUS est une explication minimale de l'incohérence. Enlever une seule contrainte suffit à la faire disparaître.

Utilité : Fournir à un utilisateur la « cause minimale » d'un conflit entre contraintes (ex : dans un système de recommandation, un diagnostic, une planification, etc.)

4.3 Définition d'un MCS

Définition 2 (MCS)**.** *Un sous-ensemble $M \subseteq C$ est un **MCS** (Minimal Correction Subset) si :*

— *$\mathsf{SAT}(C \setminus M) = \mathsf{true}$ (correction)*
— *$\forall M' \subset M,\ \mathsf{SAT}(C \setminus M') = \mathsf{false}$ (minimalité)*

Intuition : Un MCS est un ensemble minimal de contraintes qu'il faut **supprimer** pour restaurer la cohérence du système.

Utilité : En ingénierie des connaissances ou en diagnostic, les MCS indiquent quelles hypothèses ou règles doivent être retirées pour corriger un conflit.

4.4 Définition d'un MSS

Définition 3 (MSS)**.** *Un sous-ensemble $S \subseteq C$ est un **MSS** (Maximal Satisfiable Subset) si :*

- $\mathsf{SAT}(S) = \mathsf{true}$
- $\forall c \in C \setminus S,\ \mathsf{SAT}(S \cup \{c\}) = \mathsf{false}$ *(maximalité)*

Intuition : Un MSS est un ensemble maximal de contraintes parmi C qui peut encore être satisfait. Ajouter la moindre contrainte restante rend le système incohérent.

Utilité : Les MSS sont utiles pour explorer des configurations cohérentes maximales, par exemple en planification sous contraintes ou pour des suggestions cohérentes dans un système d'aide à la décision.

4.5 Liens entre MUS, MCS et MSS

Il existe une relation forte entre ces trois notions :

- Chaque **MUS** est complémentaire d'au moins un **MCS** : si M est un MCS, alors $C \setminus M$ est un MSS, et son complémentaire contient un MUS.
- Inversement, chaque **MSS** est le complément d'un MCS.
- En pratique, on peut générer les MUS à partir des MCS (et vice-versa), bien que cela puisse être coûteux.

4.6 Exemple simple

Soit $C = \{c_1, c_2, c_3\}$ avec :

$$c_1 : x > 0$$
$$c_2 : x < 5$$
$$c_3 : x < -1$$

L'ensemble C est insatisfiable. On peut identifier :

- $MUS = \{c_1, c_3\}$: ce sous-ensemble est minimalement insatisfiable.
- $MCS = \{c_3\}$: retirer c_3 restaure la cohérence.
- $MSS = \{c_1, c_2\}$: ensemble maximal cohérent.

4.7 Rôle dans l'explicabilité

Les MUS, MCS et MSS permettent :

- de produire des **explications minimales et compréhensibles** d'un conflit,
- d'identifier les contraintes en cause (MUS),

— de proposer des solutions (MCS à retirer ou MSS à conserver),
— de formaliser des algorithmes d'explication comme QuickXplain.

5 L'algorithme QuickXplain

Pour identifier un MUS de façon efficace, l'algorithme **QuickXplain** permet de calculer un sous-ensemble minimal de contraintes responsable de l'incohérence, sans énumérer tous les sous-ensembles possibles.

5.1 Principe général

QuickXplain repose sur une approche **diviser pour régner** qui :
— évite de tester toutes les combinaisons de contraintes ;
— exploite un **oracle de satisfiabilité** pour vérifier la cohérence d'un ensemble ;
— retourne un **sous-ensemble minimal** $X' \subseteq X$ tel que $X' \cup B$ est insatisfiable.

5.2 Hypothèses

— L'ensemble $C = X \cup B$ est insatisfiable.
— B est un ensemble de contraintes supposées déjà vérifiées (cohérentes).
— X contient les contraintes suspectées d'être responsables de l'incohérence.

5.3 Pseudocode de l'algorithme

Algorithm 1 QuickXplain Algorithm

Input : X (ensemble de contraintes), B (ensemble des contraintes déjà vérifiées)
Output : X' (le sous-ensemble minimal de X rendant $X' \cup B$ inconsistant)
if $\neg\mathsf{SAT}(B)$ **then**
  return $\emptyset$ {Si B est déjà inconsistant, retourner un ensemble vide}
end if
if $|X| = 1$ **then**
  return X {Si X est indécomposable, retourner X}
else
  $X_1, X_2 \leftarrow \text{split}(X)$ {Diviser X en deux sous-ensembles X_1 et X_2}
  $X'_1 \leftarrow \text{QuickXplain}(X_1,\ B \cup X_2)$ {Appliquer QuickXplain sur X_1 avec $B \cup X_2$}
  $X'_2 \leftarrow \text{QuickXplain}(X_2,\ B \cup X'_1)$ {Appliquer QuickXplain sur X_2 avec $B \cup X'_1$}
  return $X'_1 \cup X'_2$ {Retourner l'union des sous-ensembles explicatifs}
end if

5.4 Explication du fonctionnement

L'algorithme agit récursivement en réduisant le problème :
— Il commence par vérifier si B est déjà incohérent ;
— S'il ne reste qu'une seule contrainte, celle-ci constitue l'explication minimale ;
— Sinon, X est divisé en deux parties, et on cherche à expliquer l'incohérence en testant chaque moitié, en contexte avec l'autre.

5.5 Complexité

La complexité dépend du coût des appels à l'oracle de satisfiabilité. Dans le pire des cas, QuickXplain fait $O(n \log n)$ appels, ce qui est bien plus efficace qu'une énumération exhaustive.

5.6 Avantages

— Ne nécessite pas de générer tous les MUS ;
— Produit une explication minimale cohérente avec l'ensemble initial ;
— Bien adapté à l'interaction avec l'utilisateur, car les explications sont progressives et compréhensibles.

5.7 Applications

— Débogage de connaissances (ontologies, règles, systèmes experts) ;
— Planification ou configuration de systèmes ;
— Diagnostic, explication de conflits dans les systèmes décisionnels.