APPRENTISSAGE AUTOMATIQUE — SEMESTRE 4

Module 1 — Fondamentaux de l’Apprentissage Automatique

LEÇON 1

LES CONCEPTS FONDAMENTAUX DE L'APPRENTISSAGE AUTOMATIQUE

Abidjan, 2026

TABLE DES MATIÈRES

LEÇON 1 : LES CONCEPTS FONDAMENTAUX DE L'APPRENTISSAGE AUTOMATIQUE

Présentation du machine learning

L'apprentissage automatique (machine learning) est un domaine captivant. Issu de nombreuses disciplines comme la statistique, l'optimisation, l'algorithmique ou le traitement du signal, c'est un champ d'études en mutation constante qui s'est maintenant imposé dans notre société. Déjà utilisé depuis des décennies dans la reconnaissance automatique de caractères ou les filtres anti-spam, il sert maintenant à protéger contre la fraude bancaire, recommander des livres, films, ou autres produits adaptés à nos goûts, identifier les visages dans le viseur de notre appareil photo, ou traduire automatiquement des textes d'une langue vers une autre.

Dans les années à venir, le machine learning nous permettra vraisemblablement d'améliorer la sécurité routière (y compris grâce aux véhicules autonomes), la réponse d'urgence aux catastrophes naturelles, le développement de nouveaux médicaments, ou l'efficacité énergétique de nos bâtiments et industries.

Le but de ce chapitre est d'établir plus clairement ce qui relève ou non du machine learning, ainsi que des branches de ce domaine dont ce cours traitera.

Objectifs

- Définir le machine learning
- Identifier si un problème relève ou non du machine learning
- Donner des exemples de cas concrets relevant de grandes classes de problèmes de machine learning.

1.1 Qu'est-ce que le machine learning ?

Qu'est-ce qu'apprendre, comment apprend-on, et que cela signifie-t-il pour une machine ? La question de l'apprentissage fascine les spécialistes de l'informatique et des mathématiques tout autant que neurologues, pédagogues, philosophes ou artistes.

Une définition qui s'applique à un programme informatique comme à un robot, un animal de compagnie ou un être humain est celle proposée par Fabien Benureau (2015) : « L'apprentissage est une modification d'un comportement sur la base d'une expérience ».

Dans le cas d'un programme informatique, qui est celui qui nous intéresse dans cet ouvrage, on parle d'apprentissage automatique, ou *machine learning*, quand ce programme a la capacité de se modifier lui-même sans que cette modification ne soit explicitement programmée. Cette définition est celle proposée par Arthur Samuel (1959).

On peut ainsi opposer un programme classique, qui utilise une procédure et les données qu'il reçoit

en entrée pour produire en sortie des réponses, à un programme d'apprentissage automatique, qui utilise les données et les réponses afin de produire la procédure qui permet d'obtenir les secondes à partir des premières.

Exemple

Supposons qu'une entreprise veuille connaître le montant total dépensé par un client ou une cliente à partir de ses factures. Il suffit d'appliquer un algorithme classique, à savoir une simple addition : un algorithme d'apprentissage n'est pas nécessaire.

Supposons maintenant que l'on veuille utiliser ces factures pour déterminer quels produits le client est le plus susceptible d'acheter dans un mois. Bien que cela soit vraisemblablement lié, nous ne savons pas manifestement pas toutes les informations nécessaires pour ce faire. Cependant, si nous disposons de l'historique d'achat d'un grand nombre d'individus, il devient possible d'utiliser un algorithme de machine learning pour qu'il en tire un modèle prédictif nous permettant d'apporter une réponse à notre question.

Ce point de vue informatique sur l'apprentissage automatique justifie que l'on considère qu'il s'agit d'un domaine différent de celui de la statistique. Cependant, nous aurons l'occasion de voir que la frontière entre informatique et apprentissage est souvent mince. Il s'agit, fondamentalement, de modéliser un phénomène à partir de données considérées comme autant d'observations de celui-ci.

1.1.1 Pourquoi utiliser le machine learning ?

Le machine learning peut servir à résoudre des problèmes :

- que l'on ne sait pas résoudre (comme dans l'exemple de la prédiction d'achats ci-dessus) ;
- que l'on sait résoudre, mais dont on ne sait pas formaliser en termes algorithmiques comment nous les résolvons (c'est le cas par exemple de la reconnaissance d'images ou de la compréhension du langage naturel) ;
- que l'on sait résoudre, mais avec des procédures beaucoup trop gourmandes en ressources informatiques (c'est le cas par exemple de la prédiction d'interactions entre molécules de grande taille, pour lesquelles les simulations sont très lourdes).

Le machine learning est donc utilisé quand les **données sont abondantes** (relativement), mais les connaissances peu accessibles ou peu développées.

Ainsi, le machine learning peut aussi aider les humains à apprendre : les modèles créés par des algorithmes d'apprentissage peuvent révéler l'importance relative de certaines informations ou la façon dont elles interagissent entre elles pour résoudre un problème particulier. Dans l'exemple de la prédiction d'achats, comprendre le modèle peut nous permettre d'analyser quelles caractéristiques des achats passés permettent de prédire ceux à venir. Cet aspect du machine learning est très utilisé dans la recherche scientifique : quels gènes sont impliqués dans le développement d'un certain type de tumeur, et comment ?

Quelles régions d'une image cérébrale permettent de prédire un comportement ? Quels aspects

d'une molécule en font un bon médicament pour une indication particulière ? Quels aspects d'un télescope permettent d'y identifier un objet astronomique particulier ?

Ingrédients du machine learning

Le machine learning repose sur deux piliers fondamentaux :

- d'une part, **les données**, qui sont les exemples à partir desquels l'algorithme va apprendre ;
- d'autre part, **l'algorithme d'apprentissage**, qui est le procédé que l'on fait tourner sur ces données pour produire un modèle.

Ces deux piliers sont aussi importants l'un que l'autre. D'une part, aucun algorithme d'apprentissage ne pourra créer un bon modèle à partir de données qui ne sont pas pertinentes — c'est le concept *garbage in, garbage out* qui stipule qu'un algorithme d'apprentissage auquel on fournit des données de mauvaise qualité ne peut rien en faire d'autre que des prédictions de mauvaise qualité. D'autre part, un modèle appris avec un algorithme inadapté sur des données pertinentes ne pourra pas non plus être de bonne qualité.

Ce cours est consacré au deuxième de ces piliers — les algorithmes d'apprentissage. Néanmoins, il ne faut pas négliger qu'une part importante du travail de *machine learner* ou de *data scientist* est un travail d'ingénierie consistant à préparer les données afin d'éliminer les données aberrantes, gérer les données manquantes, choisir une représentation pertinente, etc.

Attention

Bien que l'usage soit souvent d'appeler les deux du même nom, il faut distinguer **l'algorithme d'apprentissage automatique** du **modèle appris** : le premier utilise les données pour produire le second, qui peut ensuite être appliqué comme un programme classique.

Un algorithme d'apprentissage permet donc de **modéliser** un phénomène à partir d'exemples. Nous considérerons ici qu'il faut pour ce faire définir et optimiser un objectif. Il peut par exemple s'agir de minimiser le nombre d'erreurs faites par le modèle sur les exemples d'apprentissage. Cet ouvrage présente en effet les algorithmes les plus classiques et les plus populaires sous cette forme.

Exemple

Voici quelques exemples de reformulation de problèmes de machine learning sous la forme d'un problème d'optimisation, formulés ici très librement :

- un site marchand peut chercher à *modéliser* des types représentatifs de clientèle, à partir des transactions passées, en maximisant la proximité entre clients et clientes affectés à un même type ;
- une compagnie automobile peut chercher à *modéliser* la trajectoire d'un véhicule dans son environnement, à partir d'enregistrements vidéo de voitures, en minimisant le nombre d'accidents ;

- des chercheurs en génétique peuvent vouloir *modéliser* l'impact d'une mutation sur une maladie, à partir de données d'un patient, en maximisant la cohérence de leur modèle avec les connaissances de l'état de l'art ;
- une banque peut vouloir *modéliser* les comportements à risque, à partir de son historique, en maximisant le taux de détection de non solvabilité.

Ainsi, le machine learning repose d'une part sur les mathématiques, et en particulier la statistique, pour ce qui est de la construction de modèles et de leur inférence à partir de données, et d'autre part sur l'informatique, pour ce qui est de la représentation des données et de l'implémentation efficace d'algorithmes d'optimisation. De plus en plus, les quantités de données disponibles imposent de faire appel à des architectures de calcul et de base de données distribuées. C'est un point important mais que nous n'abordons pas dans cet ouvrage.

Et l'intelligence artificielle, dans tout ça ?

Le machine learning peut être vu comme une branche de l'intelligence artificielle. En effet, un système incapable d'apprendre peut difficilement être considéré comme intelligent. La capacité à apprendre et à tirer parti de ses expériences est en effet essentielle à un système conçu pour s'adapter à un environnement changeant.
L'intelligence artificielle, définie comme l'ensemble des techniques mises en œuvre afin de construire des machines capables de faire preuve d'un comportement que l'on peut qualifier d'intelligent, fait aussi appel aux sciences cognitives, à la neurobiologie, à la logique, à l'électronique, à l'ingénierie et bien plus encore. Probablement parce que le terme « intelligence artificielle » stimule plus l'imagination, il est cependant de plus en plus souvent employé en lieu et place de celui d'apprentissage automatique.

Questions de société

L'essor récent de l'intelligence artificielle, en particulier à travers les progrès du machine learning, suscite de vifs débats philosophiques, éthiques et moraux. Les biais algorithmiques, et en particulier les biais de l'intelligence artificielle, sont le sujet d'inquiétudes profondes, certaines hélas bien fondées, d'autres plutôt fantasmées.

1.1.2 Exemple concret

Pour mieux comprendre la différence entre un programme classique et un programme de *machine learning,* prenons un exemple concret.

Situation :

Une entreprise de e-commerce souhaite prédire si un client va acheter un produit recommandé ou non.

a) Avec un programme classique :

Le développeur doit écrire manuellement toutes les règles :

```
     si âge > 30 et revenu > 3000€ et a déjà acheté des livres → recommander ce livre
sinon si âge < 25 et aime les jeux vidéo → recommander ce jeu
...
```

C'est très difficile, car il faudrait anticiper **toutes** les combinaisons possibles (âge, sexe, historique d'achats, heure de connexion, appareil utilisé, etc.). Le code deviendrait rapidement énorme et rigide.

b) Avec le machine learning :

On donne à l'algorithme des **milliers d'exemples** passés :

- Client A (30 ans, revenu 450000fr CFA, historique…) → a acheté
- Client B (22 ans, revenu 180000 fr CFA, historique…) → n'a pas acheté
- Client C (45 ans, revenu 6200 fr CFA, historique…) → a acheté

L'algorithme **apprend tout seul** les régularités dans les données et construit un **modèle** capable de prédire pour un nouveau client s'il est susceptible d'acheter ou non.

Avantage majeur :

Le modèle peut découvrir des relations complexes que même un expert humain n'aurait pas pensées (ex. : combinaison « heure de connexion + type d'appareil + catégorie de produits consultés récemment »).

Résumé de l'exemple :

- **Programme classique** = On donne les **règles** → l'ordinateur applique.
- **Machine Learning** = On donne les **exemples** → l'ordinateur trouve les règles.

C'est cette capacité d'**apprentissage à partir des données** qui fait toute la puissance du machine learning.

1.1.3 Caractéristiques des problèmes impossibles à résoudre algorithmiquement

Ces problèmes constituent l'une des principales raisons d'être du Machine Learning. Ils correspondent à des situations où il est impossible (ou extrêmement difficile) d'écrire un algorithme classique avec des règles explicites.

a) Problèmes impossibles à résoudre algorithmiquement

Il s'agit de cas où **aucun algorithme explicite** ne peut être écrit pour résoudre le problème.

Exemple classique :

Prédire les produits qu'un client est le plus susceptible d'acheter le mois prochain à partir de son historique

d’achats.

- Il est impossible de formaliser manuellement toutes les règles (« si le client a acheté A puis B, alors il achètera C ») car les interactions entre les variables sont extrêmement complexes.
- Le comportement humain dépend de milliers de facteurs difficiles à anticiper.

Autres exemples :

- Reconnaître un visage sur une photo
- Détecter si un email est un spam
- Traduire automatiquement un texte
- Recommander une vidéo ou un film que l’utilisateur aimera

Dans ces situations, on ne sait **pas comment** résoudre le problème avec des règles classiques. Le machine learning permet au système d’**apprendre** lui-même à partir des exemples.

b) Problèmes que l’on sait résoudre, mais que l’on ne sait pas formaliser algorithmiquement

Ce sont des tâches que les humains réalisent naturellement très bien, mais pour lesquelles il est extrêmement difficile (voire impossible) d’expliciter les règles précises.

Exemples emblématiques :

- Reconnaître un objet sur une image (reconnaissance visuelle)
- Comprendre le sens d’une phrase (compréhension du langage naturel)
- Lire une écriture manuscrite
- Interpréter le ton ou l’émotion dans une voix

Un humain sait très bien reconnaître un chat sur une photo, mais il serait incapable d’écrire un programme de règles détaillées (« si tu vois tel pixel de telle couleur à tel endroit… »). Le machine learning contourne ce problème en apprenant directement à partir d’un grand nombre d’exemples étiquetés.

c) Problèmes que l’on sait résoudre, mais dont les solutions sont trop coûteuses en ressources

Dans certains cas, une solution algorithmique existe, mais elle est **pratiquement inutilisable** en raison de sa complexité ou de son coût computationnel.

Exemples :

- Prédire les interactions entre deux grosses molécules (simulations physiques très lourdes)

- Calculer le meilleur trajet pour un véhicule en tenant compte de tous les paramètres en temps réel
- Simuler avec précision le comportement d'un nouveau médicament dans le corps humain

Le machine learning permet alors d'apprendre un **modèle approximatif** beaucoup plus rapide tout en conservant une bonne précision.

Synthèse : Quand faut-il privilégier le Machine Learning ?

On recourt au machine learning quand :

- Les **données sont abondantes** (ou peuvent l'être)
- Les **connaissances expertes** sont limitées ou difficiles à formaliser
- Le problème est trop complexe pour des règles manuelles
- On cherche à gagner en **vitesse** ou en **échelle**

Le machine learning excelle particulièrement dans les domaines où les **données sont riches** mais la **théorie est pauvre**.

1.1.4 Les deux piliers du Machine Learning : Données et Algorithmes

Le machine learning repose sur **deux piliers fondamentaux**, indissociables et d'égale importance :

a) Les Données

Les données constituent les **exemples d'expérience** à partir desquels le système va apprendre. Elles sont le carburant du machine learning.

- Elles peuvent être des images, des textes, des tableaux de chiffres, des enregistrements audio, des vidéos, etc.
- La qualité, la quantité et la représentativité des données sont **cruciales**.
- Un adage bien connu résume parfaitement cet enjeu : **« Garbage In, Garbage Out »** (si les données d'entrée sont mauvaises, les résultats seront mauvais).

Même le meilleur algorithme du monde ne pourra pas produire un bon modèle à partir de données :

- Bruitées,
- Biaisées,
- Incomplètes,
- Non représentatives du problème réel.

C’est pourquoi une part très importante du travail d’un *data scientist* ou *machine learner* consiste à **préparer et nettoyer** les données (nettoyage, traitement des valeurs manquantes, ingénierie des caractéristiques, etc.).

b) L’Algorithme d’Apprentissage

L’algorithme d’apprentissage est le **procédé mathématique et informatique** qui, à partir des données, va construire un modèle.

- Il explore les données pour identifier des régularités, des patterns ou des relations.
- Il optimise un objectif (par exemple : minimiser le nombre d’erreurs de prédiction).
- Il produit en sortie un **modèle** (un ensemble de règles ou de paramètres appris).

Attention : importante distinction

- **Algorithme d’apprentissage** = la méthode d’entraînement (ex. : Régression Logistique, Arbres de Décision, Réseaux de Neurones, etc.).
- **Modèle** = le résultat de cet apprentissage (un programme prêt à être utilisé pour faire des prédictions sur de nouvelles données).

On peut voir l’algorithme comme la **recette de cuisine**, et le modèle comme le **plat final** obtenu après avoir suivi cette recette avec les ingrédients (données).

Synthèse visuelle

Pilier	Rôle	Importance	Exemples
Données	Fournir les exemples d’apprentissage	Qualité > Quantité	Images étiquetées, historiques clients
Algorithmes	Apprendre et construire le modèle	Choix adapté au type de problème	Régression, SVM, Random Forest…

Remarque importante

Ce cours se concentre principalement sur le **deuxième pilier** : les algorithmes d’apprentissage. Cependant, rappelez-vous toujours qu’un excellent algorithme sur de mauvaises données donnera de mauvais résultats.

1.1.5 Distinction Importante: Algorithme d’Apprentissage vs Modèle

Une confusion très fréquente, même chez les débutants comme chez certains praticiens, consiste à utiliser indistinctement les termes **« algorithme »** et **« modèle »**. Pourtant, il s’agit de deux concepts bien

distincts.

Définition claire :

- **Algorithme d'apprentissage** :
  C'est la **méthode**, la **procédure** ou la **recette** qui permet d'apprendre à partir des données. Il s'agit d'un ensemble d'instructions mathématiques et informatiques qui analyse les données et optimise des paramètres.
- **Modèle** :
  C'est le **résultat** de cet apprentissage. C'est un objet concret (ensemble de paramètres, de règles ou de poids) qui peut ensuite être utilisé comme un programme classique pour faire des prédictions sur de **nouvelles données**.

Analogies pour mieux comprendre :

Concept	Analogie Culinaire	Analogie Scolaire	Rôle dans le Machine Learning
Algorithme	La recette de cuisine	Le professeur / la méthode pédagogique	Apprend à partir des données
Modèle	Le plat cuisiné	Les connaissances acquises par l'élève	Prédit sur de nouvelles données
Données	Les ingrédients	Les exercices et leçons	Exemples d'apprentissage

Exemple concret :
Supposons que nous voulions prédire le prix d'une maison :

1. On choisit un **algorithme** (ex. : Régression Linéaire ou Arbres de Décision).
2. On lui donne les données d'entraînement (surface, nombre de pièces, localisation, etc. + prix réels).
3. L'algorithme **apprend** et produit un **modèle**.
4. Ce modèle peut ensuite être utilisé des milliers de fois pour estimer le prix de nouvelles maisons jamais vues auparavant.

Important :
Une fois entraîné, le **modèle** se comporte comme un programme classique : on lui donne une entrée → il donne une sortie, sans réapprendre à chaque fois.

En résumé

- L'**algorithme** est utilisé **pendant la phase d'apprentissage** (entraînement).
- Le **modèle** est utilisé **après l'apprentissage** (inférence / prédiction).

- On entraîne un algorithme **une fois** (ou régulièrement) pour obtenir un modèle que l'on déploie ensuite des millions de fois.

Cette distinction est fondamentale : un bon algorithme sur de mauvaises données donne un mauvais modèle. Un excellent algorithme mal configuré donne aussi un mauvais modèle.

1.1.6 Machine Learning et Intelligence Artificielle

Le **machine learning** est étroitement lié à l'**intelligence artificielle (IA)**, mais les deux concepts ne sont pas synonymes.

Définition de l'Intelligence Artificielle

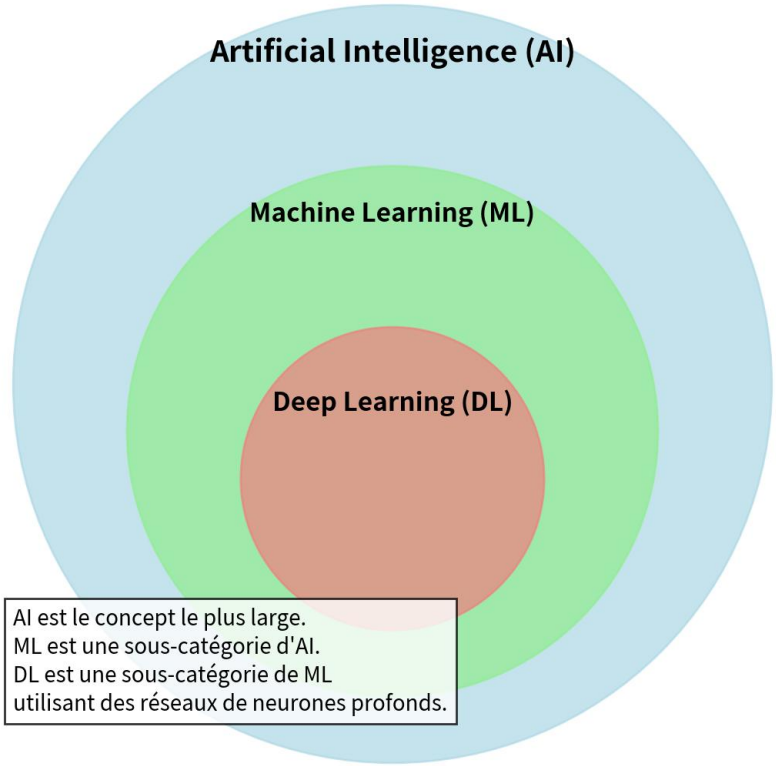
L'intelligence artificielle désigne **l'ensemble des techniques** permettant de construire des machines capables d'accomplir des tâches qui, jusqu'à présent, requéraient l'intelligence humaine. Cela inclut non seulement l'apprentissage, mais aussi le raisonnement, la planification, la perception, la compréhension du langage, la créativité, etc.

Position du Machine Learning dans l'IA

Le machine learning est aujourd'hui **la branche la plus importante et la plus performante** de l'intelligence artificielle.

On peut représenter cette relation de la façon suivante :

Relations entre IA, Machine Learning et Deep Learning



Pourquoi le machine learning est-il considéré comme une branche de l'IA ?

Parce qu'un système qui ne peut **pas apprendre** de son expérience a du mal à être qualifié d'« intelligent » dans un environnement complexe et changeant. La capacité d'adaptation par l'apprentissage est une caractéristique fondamentale de l'intelligence.

Évolution des termes

- Historiquement, l'IA englobait de nombreuses approches (systèmes experts, algorithmes de recherche, logique floue…).
- Depuis les années 2010, grâce aux progrès spectaculaires du machine learning (et particulièrement du **deep learning**), le terme « Intelligence Artificielle » est de plus en plus utilisé dans le langage courant pour désigner en réalité des systèmes de **machine learning**.
- Cette substitution de termes est parfois abusive, mais elle reflète bien l'importance actuelle du machine learning au sein de l'IA.

En résumé

- **Intelligence Artificielle** = concept large (ambition générale).

- **Machine Learning** = approche dominante actuelle de l'IA, basée sur l'apprentissage à partir de données.
- **Deep Learning** = sous-domaine du machine learning utilisant des réseaux de neurones profonds.

Le machine learning est donc **un moyen** d'atteindre l'intelligence artificielle, et non l'IA elle-même.

1.1.7 Enjeux Sociétaux et Éthiques

L'essor fulgurant du machine learning et de l'intelligence artificielle ne soulève pas uniquement des questions techniques. Il pose également de **profonds défis philosophiques, éthiques, sociaux et juridiques** qui concernent notre société tout entière.

Principaux enjeux actuels

a) Les Biais Algorithmiques

Les modèles de machine learning apprennent à partir de données historiques. Si ces données contiennent des biais (discrimination liée au genre, à l'origine ethnique, à l'âge, etc.), le modèle risque de les reproduire, voire de les amplifier.

Exemples célèbres :

- Systèmes de reconnaissance faciale moins performants sur certaines couleurs de peau
- Outils de recrutement favorisant inconsciemment un profil masculin
- Algorithmes judiciaires surestimant le risque de récidive pour certaines catégories de population

b) La Transparence et l'Explicabilité

De nombreux modèles (notamment les réseaux de neurones profonds) sont considérés comme des « boîtes noires ». Il est souvent difficile de comprendre pourquoi ils ont pris telle ou telle décision. Cela pose problème dans des domaines sensibles comme la santé, la justice ou le crédit bancaire.

c) La Vie Privée et la Protection des Données

L'entraînement des modèles nécessite souvent d'énormes quantités de données personnelles. Cela soulève des questions sur le consentement, la surveillance de masse et le respect de la vie privée (RGPD en Europe).

d) L'Impact sur l'Emploi

L'automatisation de tâches cognitives (traduction, analyse d'images, rédaction, diagnostic médical, etc.) pourrait transformer profondément le marché du travail.

e) La Responsabilité

En cas d'erreur grave (accident d'une voiture autonome, diagnostic médical erroné, etc.), qui est

responsable ? Le développeur ? L'entreprise ? L'utilisateur ? L'algorithme lui-même ?

f) La Désinformation et les Contenus Générés

Les progrès du machine learning facilitent la création de *deepfakes* (fausses vidéos ou fausses voix très réalistes), ce qui menace la confiance dans l'information.

Point de vue équilibré

Si certains risques sont bien réels et méritent une vigilance forte, d'autres inquiétudes sont parfois exagérées ou fantasmées. Le machine learning est aussi un outil puissant pour le bien commun :

- Accélération de la recherche médicale
- Lutte contre le changement climatique
- Amélioration de l'accessibilité pour les personnes en situation de handicap
- Détection précoce des catastrophes naturelles

En résumé

Le machine learning n'est ni intrinsèquement bon ni mauvais : c'est un outil extrêmement puissant dont les effets dépendent de la façon dont nous le concevons, l'entraînons et le déployons.

Il est donc essentiel que les futurs ingénieurs, data scientists et décideurs maîtrisent non seulement la technique, mais aussi les **implications éthiques** de leurs choix.

1.2 Types de problèmes de machine learning

Le machine learning est un champ assez vaste, et nous dressons dans cette section une liste des plus grandes classes de problèmes auxquels il s'intéresse.

1.2.1 Apprentissage supervisé

L'apprentissage supervisé est peut-être le type de problèmes de machine learning le plus facile à appréhender : son but est d'apprendre à faire des prédictions, à partir d'une liste d'exemples étiquetés, c'est-à-dire accompagnés de la valeur à prédire (voir figure 1.1). Les étiquettes servent de « professeur » et supervisent l'apprentissage de l'algorithme.

Définition 1.1 (Apprentissage supervisé) On appelle apprentissage supervisé la branche du machine learning qui s'intéresse aux problèmes pouvant être formalisés de la façon suivante : étant données n observations $\{x^i\}_{i=1,\dots,n}$ décrites dans un espace $\mathcal{X}$, et leurs étiquettes $\{y^i\}_{i=1,\dots,n}$ décrites dans un espace $\mathcal{Y}$, on suppose que les étiquettes peuvent être obtenues à partir des observations grâce à une fonction $\phi: \mathcal{X} \to \mathcal{Y}$ fixe et inconnue : $y^i = \phi(x^i) + \epsilon_i$, où ϵ_i est un bruit aléatoire. Il s'agit alors d'utiliser les données pour déterminer une fonction $f: \mathcal{X} \to \mathcal{Y}$ telle que, pour tout couple $(x^i, y^i) \in \mathcal{X} \times \mathcal{Y}$, $f(x^i) \approx \phi(x^i)$.

D'un point de vue probabiliste, on suppose que les données étiquetées (x^i, y^i) sont autant de

réalisations d'un même couple de variables aléatoires (X, Y), qui vérifie $Y = \phi(X) + \epsilon$, avec ϵ une variable aléatoire représentant un bruit.

L'espace sur lequel sont définies les données est le plus souvent $\mathcal{X} = \mathbb{R}^p$. Nous verrons cependant aussi d'autres types de représentations, comme des variables binaires, discrètes, catégoriques, voire des chaînes de caractères ou des graphes.

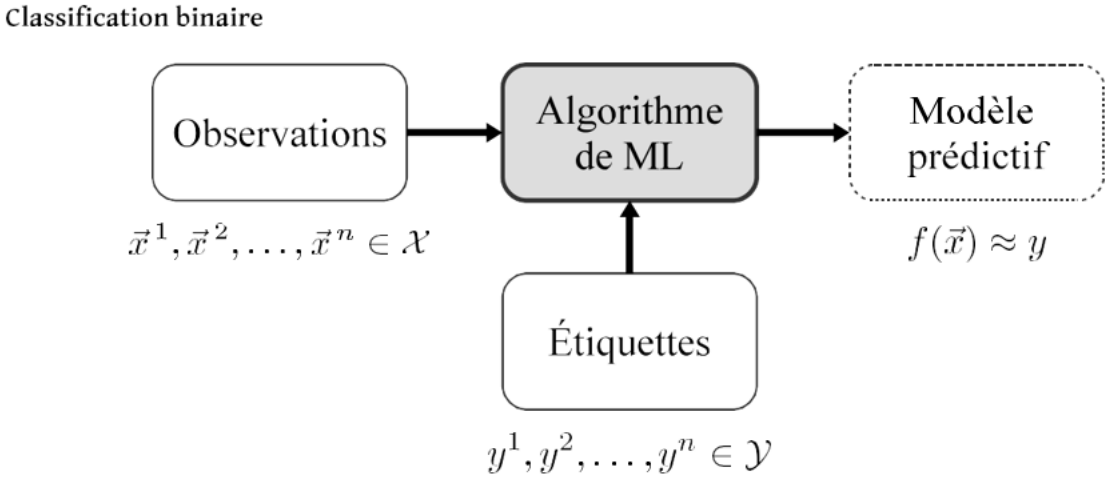


FIGURE 1.1 – Apprentissage supervisé.

Dans le cas où les étiquettes sont binaires, elles indiquent l'appartenance à une classe. On parle alors de classification binaire.

Définition 1.2 (Classification binaire) Un problème d'apprentissage supervisé dans lequel l'espace des étiquettes est binaire, autrement dit $\mathcal{Y} = \{0,1\}$, est appelé un problème de classification binaire.

Exemple

Voici quelques exemples de problèmes de classification binaire :

- Identifier si un email est un spam ou non ;
- Identifier si un tableau a été peint par Picasso ou non ;
- Identifier si une image contient ou non une girafe ;
- Identifier si une molécule peut ou non traiter la dépression ;
- Identifier si une transaction financière est frauduleuse ou non.

Classification multi-classe

Dans le cas où les étiquettes sont discrètes, et correspondent donc à plusieurs classes, on parle de classification multi-classe.

Définition 1.3 (Classification multi-classe) Un problème d'apprentissage supervisé dans lequel l'espace des étiquettes est discret et fini, autrement dit $\mathcal{Y} = \{1,2,\dots,C\}$ est appelé un problème de classification multi-classe. C est le nombre de classes.

Exemple

Voici quelques exemples de problèmes de classification multi-classe :

- Identifier en quelle langue un texte est écrit ;
- Identifier lequel des 10 chiffres arabes est un chiffre manuscrit ;
- Identifier l'expression d'un visage parmi une liste prédéfinie de possibilités (colère, tristesse, joie, etc.) ;
- Identifier à quelle espèce appartient une plante ;
- Identifier les objets présents sur une photographie.

Régression

Dans le cas où les étiquettes sont à valeurs réelles, on parle de régression.

Définition 1.4 (Régression) Un problème d'apprentissage supervisé dans lequel l'espace des étiquettes est $\mathcal{Y} = \mathbb{R}$ est appelé un problème de régression.

Exemple

Voici quelques exemples de problèmes de régression :

- Prédire le nombre de clics sur un lien ;
- Prédire le nombre d'utilisateurs et utilisatrices d'un service en ligne à un moment donné ;
- Prédire le prix d'une action en bourse ;
- Prédire l'affinité de liaison entre deux molécules ;
- Prédire le rendement d'un plant de maïs.

Régression structurée

Dans le cas où l'espace des étiquettes est un espace structuré plus complexe que ceux évoqués précédemment, on parle de régression structurée – en anglais, *structured regression*, ou *structured output prediction*. Il peut par exemple s'agir de prédire des vecteurs, des images, des graphes, ou des séquences. La régression structurée permet de formaliser de nombreux problèmes, comme ceux de la traduction automatique ou de la reconnaissance vocale (text-to-speech et speech-to-text, par exemple). Ce cas dépasse cependant le cadre du présent ouvrage.

et nous nous concentrerons sur les problèmes de classification binaire et multi-classe, ainsi que de régression classique.

L'apprentissage supervisé est le sujet principal de cet ouvrage, et sera traité du chapitre 2 au chapitre 9.

1.2.2 Apprentissage non supervisé

Dans le cadre de l'apprentissage non supervisé, les données ne sont pas étiquetées. Il s'agit alors de modéliser les observations pour mieux les comprendre (voir figure 1.2).

Définition 1.5 (Apprentissage non supervisé) On appelle apprentissage non supervisé la branche du machine learning qui s'intéresse aux problèmes pouvant être formalisés de la façon suivante : étant données n observations $\{x^i\}_{i=1,\dots,n}$ décrites dans un espace $\mathcal{X}$, il s'agit d'apprendre une nouvelle représentation de ces observations, considérée comme plus informative

FIGURE 1.2 – Apprentissage non supervisé.

Cette définition est très vague, et sera certainement plus claire sur les exemples qui suivent.

Clustering

Tout d'abord, le *clustering*, ou partitionnement, consiste à identifier des groupes dans les données (voir figure 1.3). Cela permet de comprendre leurs caractéristiques générales, et éventuellement d'inférer les propriétés d'une observation en fonction du groupe auquel elle appartient.

Définition 1.6 (Partitionnement) On appelle partitionnement ou clustering un problème d'apprentissage non supervisé pouvant être formalisé comme la recherche d'une partition $\{C_k\}_{k=1\dots K}$ des n observations $\{x^i\}_{i=1,\dots,n}$. Cette partition doit être pertinente au vu d'un ou plusieurs critères à préciser. Chaque observation est maintenant représentée par la partie (*cluster*) à laquelle elle appartient.

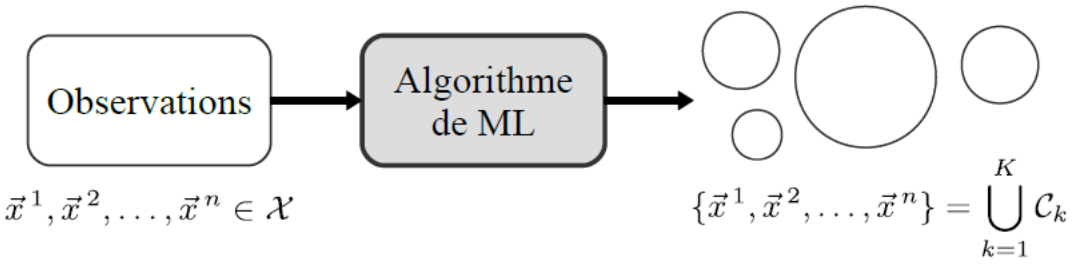


FIGURE 1.3 – Partitionnement des données, ou clustering.

Exemple

Voici quelques exemples de problèmes de partitionnement :

- La segmentation de marché consiste à identifier des groupes d'usagers ou de clients ayant un comportement similaire. Cela permet de mieux comprendre leur profil, et cibler une campagne de publicité, des contenus ou des actions spécifiquement vers certains groupes.
- Identifier des groupes de documents ayant un sujet similaire, sans les avoir au préalable étiquetés par sujet. Cela permet d'organiser de larges banques de textes.
- La compression d'image peut être formulée comme un problème de partitionnement consistant à regrouper des pixels similaires pour ensuite les représenter plus efficacement.
- La segmentation d'image consiste à identifier les pixels d'une image appartenant à la même région.
- Identifier des groupes parmi les patients présentant les mêmes symptômes permet d'identifier des sous-types d'une maladie, qui pourront être traités différemment.

Ce sujet est traité en détail au chapitre 12.

Réduction de dimension

La réduction de dimension est une autre famille importante de problèmes d'apprentissage non supervisé. Il s'agit de trouver une représentation des données dans un espace de dimension plus faible que celle de l'espace $\mathcal{X}$ dans lequel elles sont représentées à l'origine (voir figure 1.4). Cela permet de réduire le temps de calcul et l'espace mémoire nécessaire au stockage des données, mais aussi souvent d'améliorer les performances d'un algorithme d'apprentissage supervisé entraîné par la suite sur ces données.

Définition 1.7 (Réduction de dimension) On appelle réduction de dimension un problème d'apprentissage non supervisé pouvant être formalisé comme la recherche d'un espace $\mathcal{Z}$ de dimension plus faible que l'espace $\mathcal{X}$ dans lequel sont représentées n observations $\{x^i\}_{i=1,\dots,n}$. Les projections $\{z^i\}_{i=1,\dots,n}$ des données sur $\mathcal{Z}$ doivent vérifier certaines propriétés à préciser. Chaque observation est maintenant représentée par sa projection sur $\mathcal{Z}$.

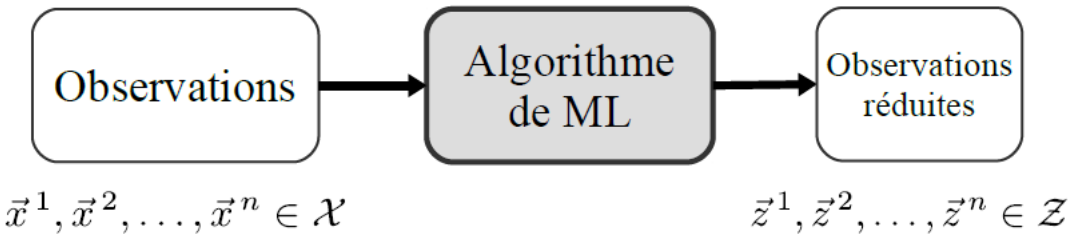


FIGURE 1.4 – Réduction de dimension.

Remarque

Certaines méthodes de réduction de dimension sont supervisées : il s'agit alors de trouver la représentation la plus pertinente pour prédire une étiquette donnée.

Nous traiterons de la réduction de dimension au chapitre 11

Estimation de densité

Enfin, une grande famille de problèmes d'apprentissage non supervisé est en fait un problème traditionnel d'inférence statistique : il s'agit de supposer que le jeu de données est un échantillon d'une variable aléatoire X, puis d'estimer la loi de probabilité de X. Les observations sont maintenant représentées par cette loi de probabilité. L'appendice B aborde brièvement ce sujet.

1.2.3 Apprentissage semi-supervisé

Comme on peut s'en douter, l'apprentissage semi-supervisé consiste à apprendre des étiquettes à partir d'un jeu de données partiellement étiqueté. Le premier avantage de cette approche est qu'elle permet d'éviter d'avoir à étiqueter l'intégralité des exemples d'apprentissage, ce qui est pertinent quand il est facile d'accumuler des données mais que leur étiquetage requiert une certaine quantité de travail humain.

Prenons par exemple la classification d'images : il est facile d'obtenir une banque de données contenant des centaines de milliers d'images, mais avoir pour chacune d'entre elles l'étiquette qui nous intéresse peut requérir énormément de travail. De plus, les étiquettes données par des humains sont susceptibles de reproduire des biais humains, qu'un algorithme entièrement supervisé reproduira à son tour. L'apprentissage semi-supervisé permet parfois d'éviter cet écueil.

1.2.4 Apprentissage par renforcement

Dans le cadre de l'apprentissage par renforcement, le système d'apprentissage peut interagir avec son environnement et accomplir des actions. En retour de ces actions, il obtient une récompense, qui peut être positive si l'action était un bon choix, ou négative dans le cas contraire. La récompense peut parfois venir après une longue suite d'actions ; c'est le cas par exemple pour un système apprenant à jouer au go ou aux échecs. Ainsi, l'apprentissage consiste dans ce cas à définir une politique, c'est-à-dire une stratégie permettant d'obtenir systématiquement la meilleure récompense possible.

Les applications principales de l'apprentissage par renforcement se trouvent dans les jeux (échecs, go, etc.) et la robotique.

1.3 Ressources pratiques

1.3.1 Implémentations logicielles

De nombreux logiciels et librairies open source permettent de mettre en œuvre des algorithmes de machine learning. Nous en citons ici quelques-uns :

- Les exemples de ce livre ont été écrits en **Python**, grâce à la très utilisée librairie **scikit-learn** (http://scikit-learn.org) dont le développement, soutenu entre autres par Inria et Télécom ParisTech, a commencé en 2007.
- De nombreux outils de machine learning sont implémentés en **R**, et recensés sur la page http://cran.r-project.org/web/views/MachineLearning.html
- **Weka** (Waikato environment for knowledge analysis, https://www.cs.waikato.ac.nz/ml/weka/) est une suite d'outils de machine learning écrits en Java et dont le développement a commencé en 1993.
- **Shogun** (http://www.shogun-toolbox.org/) interface de nombreux langages, et en particulier Python, Octave, R, et C#. Shogun, ainsi nommée d'après ses fondateurs Søren Sonnenburg et Gunnar Rätsch, a vu le jour en 1999.

De nombreux outils spécialisés pour l'apprentissage profond et le calcul sur des architectures distribuées ont vu le jour ces dernières années. Parmi eux, **TensorFlow** (https://www.tensorflow.org) implémente de nombreux autres algorithmes de machine learning.

1.3.2 Jeux de données

De nombreux jeux de données sont disponibles publiquement et permettent de se faire la main ou de tester de nouveaux algorithmes de machine learning. Parmi les ressources incontournables, citons :

- Le répertoire de l'Université de Californie à Irvine, **UCI Repository** (https://archive.ics.uci.edu/ml/index.php)
- Les ressources listées sur **KD Nuggets** : https://www.kdnuggets.com/datasets/index.html
- La plateforme de compétitions en sciences des données **Kaggle** (https://www.kaggle.com/)

1.4 Notations

Autant que faire se peut, nous utilisons dans cet ouvrage les notations suivantes :

- les lettres minuscules (x) représentent un scalaire ;
- les lettres minuscules surmontées d'une flèche ($\vec{x}$) représentent un vecteur ;
- les lettres majuscules (X) représentent une matrice, un événement ou une variable aléatoire ;
- les lettres calligraphiées ($\mathcal{X}$) représentent un ensemble ou un espace ;
- les indices correspondent à une variable tandis que les exposants correspondent à une observation : x_j^i est la j-ème variable de la i-ème observation, et correspond à l'entrée $X_{i,j}$ de la matrice X ;
- n est un nombre d'observations, p un nombre de variables, C un nombre de classes ;
- $[a]_+$ représente la partie positive de $a \in \mathbb{R}$, autrement dit $\max(0, a)$;
- $P(A)$ représente la probabilité de l'événement A ;

- $E[X]$ représente l'espérance de la variable aléatoire X ;
- $V[X]$ représente la variance de la variable aléatoire X ;
- 1_A est la fonction indicatrice

$$1_A = \begin{cases} 1 & \text{si } A \text{ est vraie} \\ 0 & \text{sinon;} \end{cases}$$

- $\langle \cdot, \cdot \rangle$ représente le produit scalaire sur $\mathbb{R}^p$;
- $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ représente le produit scalaire sur $\mathcal{H}$;
- $M \succeq 0$ signifie que M est une matrice symétrique semi-définie positive.

Points clefs

- Un algorithme de machine learning est un algorithme qui apprend un modèle à partir d'exemples, par le biais d'un problème d'optimisation.
- On utilise le machine learning lorsqu'il est difficile ou impossible de définir les instructions explicites à donner à un ordinateur pour résoudre un problème, mais que l'on dispose de nombreux exemples illustratifs.
- Les algorithmes de machine learning peuvent être divisés selon la nature du problème qu'ils cherchent à résoudre, en apprentissage supervisé, non supervisé, semi-supervisé, et par renforcement.

Bibliographie

1. *Bakır, G., Hofmann, T., Schölkopf, B., Smola, A. J., Taskar, B., et Vishwanathan, S. V. N. (2007). Predicting Structured Data. MIT Press, Cambridge, MA. https://mitpress.mit.edu/books/predicting-structured-data.*
2. *Benureau, F. (2015). Self-Exploration of Sensorimotor Spaces in Robots. Thèse de doctorat, Université de Bordeaux.*
3. *Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. IBM Journal of Research and Development, 44(1.2):206–226.*
4. *Scott, D. W. (1992). Multivariate density estimation. Wiley, New York.*
5. *Sutton, R. S. et Barto A. G. (2018). Reinforcement Learning : An Introduction. MIT Press, Cambridge, MA. http://incompleteideas.net/book/the-book-2nd.html.*