Introduction au Machine Learning

2e édition

Chloé-Agathe Azencott
Maîtresse de conférences au CBIO (Centre de bio-informatique)
de Mines Paris, de l'Institut Curie et de l'INSERM

DUNOD

Illustration de couverture : © vvvita - Shutterstock

11 rue Paul Bert, 92240 Malakoff
www.dunod.com
ISBN 978-2-10-083476-1

TABLE DES MATIÈRES

AVANT-PROPOS

Le *machine learning* (apprentissage automatique) est au cœur de la science des données et de l'intelligence artificielle. Que l'on parle de transformation numérique des entreprises, de Big Data ou de stratégie nationale ou européenne, le machine learning est devenu incontournable. Ses applications sont nombreuses et variées, allant des moteurs de recherche et de la reconnaissance de caractères à la recherche en génomique, l'analyse des réseaux sociaux, la publicité ciblée, la vision par ordinateur, la traduction automatique ou encore le trading algorithmique.

À l'intersection des statistiques et de l'informatique, le machine learning se préoccupe de la modélisation des données. Les grands principes de ce domaine ont émergé des statistiques fréquentistes ou bayésiennes, de l'intelligence artificielle ou encore du traitement du signal. Dans ce livre, nous considérons que le machine learning est la science de l'apprentissage automatique d'une fonction prédictive à partir d'un jeu d'observations de données étiquetées ou non.

Ce livre se veut une introduction aux concepts et algorithmes qui fondent le machine learning, et en propose une vision centrée sur la minimisation d'un risque empirique par rapport à une classe donnée de fonctions de prédictions.

Objectifs pédagogiques : Le but de ce livre est de vous accompagner dans votre découverte du machine learning et de vous fournir les outils nécessaires à :

1. identifier les problèmes qui peuvent être résolus par des approches de machine learning ;
2. formaliser ces problèmes en termes de machine learning ;
3. identifier les algorithmes classiques les plus appropriés pour ces problèmes et les mettre en œuvre ;
4. implémenter ces algorithmes par vous-même afin d'en comprendre les tenants et aboutissants ;
5. évaluer et comparer de la manière la plus objective possible les performances de plusieurs algorithmes de machine learning pour une application particulière.

Public visé : Ce livre s'adresse à des étudiantes ou étudiants en informatique ou maths appliquées, niveau L3 ou M1 (ou deuxième année d'école d'ingénieur), qui cherchent à comprendre les fondements des principaux algorithmes utilisés en machine learning. Il se base sur mes cours à CentraleSupélec, à Mines Paris et sur OpenClassrooms et suppose les prérequis suivants :

- algèbre linéaire (inversion de matrice, théorème spectral, décomposition en valeurs propres et vecteurs propres) ;
- notions de probabilités (variable aléatoire, distributions, théorème de Bayes).

Plan du livre : Ce livre commence par une vue d'ensemble du machine learning et des différents types de problèmes qu'il permet de résoudre. Il présente comment ces problèmes peuvent être formulés mathématiquement comme des problèmes d'optimisation (chapitre 1) et pose en annexe les bases d'optimisation convexe nécessaires à la compréhension des algorithmes présentés par la suite. La majeure partie de ce livre concerne les problèmes d'apprentissage supervisé ; le chapitre 2 détaille plus particulièrement leur formulation et introduit les notions d'espace des hypothèses, de risque et perte, et de généralisation. Avant d'étudier les algorithmes d'apprentissage supervisé les plus classiques et fréquemment utilisés, il est essentiel de comprendre comment évaluer un modèle sur un jeu de données, et de savoir sélectionner le meilleur modèle parmi plusieurs possibilités, ce qui est le sujet du chapitre 3.

Il est enfin pertinent à ce stade d'aborder l'entraînement de modèles prédictifs supervisés. Le livre aborde tout d'abord les modèles paramétriques, dans lesquels la fonction modélisant la distribution des données ou permettant de faire des prédictions a une forme analytique explicite. Des éléments d'apprentissage bayésien, présentés dans le chapitre 4, complémenté par une annexe qui pose les bases de l'estimation ponctuelle, sont ensuite appliqués à des modèles d'apprentissage supervisé paramétriques (chapitre 5). Le chapitre 6 présente les variantes régularisées de ces algorithmes. Enfin, le chapitre 7 sur les réseaux de neurones propose de construire des modèles paramétriques beaucoup plus complexes et d'aborder les bases du deep learning.

Le livre aborde ensuite les modèles non paramétriques, à commencer par une des plus intuitives de ces approches, la méthode des plus proches voisins (chapitre 8). Suivront ensuite les approches à base d'arbres de décision, puis les méthodes à ensemble qui permettront d'introduire deux des algorithmes de machine learning supervisé les plus puissants à l'heure actuelle : les forêts aléatoires et le boosting de gradient (chapitre 9). Le chapitre 10 sur les méthodes à noyaux, introduites grâce aux machines à vecteurs de support, permettra de voir comment construire des modèles non linéaires via des modèles linéaires dans un espace de redescription des données.

Enfin, le chapitre 11 présentera la réduction de dimension, supervisée ou non-supervisée, et le chapitre 12 traitera d'un des problèmes les plus importants en apprentissage non supervisé : le clustering.

Chaque chapitre sera conclu par quelques exercices.

Comment lire ce livre : Ce livre a été conçu pour être lu linéairement. Cependant, après les trois premiers chapitres, il vous sera possible de lire les suivants dans l'ordre qui vous conviendra, à l'exception du chapitre 6, qui a été écrit dans la continuité du chapitre 5. De manière générale, des références vers les sections d'autres chapitres apparaîtront si nécessaire.

Remerciements : Cet ouvrage n'aurait pas vu le jour sans Jean-Philippe Vert, qui m'a fait découvrir le machine learning, avec qui j'ai enseigné et pratiqué cette discipline pendant plusieurs années, et qui m'a fait, enfin, l'honneur d'une relecture attentive.

Ce livre doit beaucoup aux personnes qui m'ont enseigné le machine learning, et plus particulièrement Pierre Baldi, Padhraic Smyth, et Max Welling ; à celles avec qui je l'ai pratiqué, notamment les membres du Baldi Lab à UC Irvine, du MLCB et du département d'inférence empirique de l'Institut Max Planck à Tübingen, et du CBIO à Mines Paris, et bien d'autres encore qu'il serait difficile de nommer exhaustivement ici ; à celles avec qui je l'ai enseigné, Karsten Borgwardt, Yannis Chaouche, Frédéric Guyon, Fabien Moutarde, mais aussi Judith Abecassis, Eugene Belilovsky, Joseph Boyd, Peter Naylor, Benoît Playe, Mihir Sahasrabudhe, Jiaqian Yu, et Luc Bertrand ; et enfin à celles auxquelles je l'ai enseigné, en particulier dans le cadre du cours Data Mining in der Bioinformatik de l'université de Tübingen (ma toute première tentative d'enseignement des méthodes à noyaux en 2012 !) et les élèves de Centrale qui ont essuyé les plâtres de la première version de ce cours à l'automne 2015.

Mes cours sont le résultat de nombreuses sources d'inspirations accumulées au cours des années. Je remercie tout particulièrement Ethem Alpaydin, David Barber, Christopher M. Bishop, Stephen Boyd, Hal Daumé III, Jerome Friedman, Trevor Hastie, Tom Mitchell, Bernhard Schölkopf, Alex Smola, Robert Tibshirani, Lieven Vandenberghe, et Alice Zhang pour leurs ouvrages.

Parce que tout serait différent sans scikit-learn, je remercie chaleureusement tous ses core-devs, et en particulier Alexandre Gramfort, Olivier Grisel, Gaël Varoquaux et Nelle Varoquaux.

Je remercie aussi Matthew Blaschko, qui m'a poussée à l'eau, et Nikos Paragios, qui l'y a encouragé.

Parce que je n'aurais pas pu écrire ce livre seule, merci à Jean-Luc Blanc des éditions Dunod, et à celles et ceux qui ont relu tout ou partie de cet ouvrage, en particulier Judith Abecassis, Luc Bertrand, Caroline Petitjean, Denis Rousselle, Erwan Scornet.

La relecture attentive de Jean-Marie Monier, ainsi que les commentaires d'Antoine Brault, ont permis d'éliminer de nombreuses coquilles et approximations de la première version de ce texte.

Merci à Alix Deleporte, enfin, pour ses relectures et son soutien.

La deuxième édition de cet ouvrage s'est particulièrement enrichie de mes enseignements à Mines Paris, de mes interactions avec les élèves des promotions 2019 et 2020, et des cours donnés avec, en plus de certaines des personnes sus-citées, Jesus Bujalance Martin, Lucia Clarotto, Nicolas Desassis, Joseph Gesnouin, Vivien Goepp, Arthur Imbert, Tristan Lazard, Matthieu Najm, Asma Nouira, Thibaud Martinez, Romain Ménégaux, et Daniel Zyss. Merci de vos questions, de vos commentaires, et de ce que vous m'avez appris. Merci aussi à Stéphane Canu, Laure Reboul, et Joseph Salmon pour les documents mis en ligne, sur lesquels je me suis appuyée pour compléter cette édition.

1 PRÉSENTATION DU MACHINE LEARNING

INTRODUCTION

Le 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 cet ouvrage 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 donnée 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 n'avons 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 inférence statistique et apprentissage est souvent mince. Il s'agit ici, 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 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 ? Quelles caractéristiques d'une molécule en font un bon médicament pour une indication particulière ? Quels aspects d'une image de 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 duquel l'algorithme va apprendre ;
- d'autre part, l'*algorithme d'apprentissage*, qui est la procédure que l'on fait tourner sur ces données pour produire un modèle. On appelle *entraînement* le fait de faire tourner un algorithme d'apprentissage sur un jeu de données.

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 pourra 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 être de bonne qualité.

Cet ouvrage 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.

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 considérons 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. La suite de cet ouvrage devrait vous éclairer sur la formalisation mathématique de ces problèmes, 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 chercheuses en génétique peuvent vouloir *modéliser* l'impact d'une mutation sur une maladie, à partir de données 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. Si ce livre n'aborde pas les questions de société liées à l'utilisation de données massives, de leur acquisition à leur interprétation, il serait malhonnête de prétendre pouvoir détacher les mathématiques et l'informatique du contexte de leur utilisation. Les ressources sur ces questions sont nombreuses, de *Algorithmes : la bombe à retardement* de Cathy O'Neil (édition Les Arènes) aux publications du AI Now Institute

(https://ainowinstitute.org) pour ne citer que deux exemples. Je ne peux que vous encourager, avant d'utiliser les outils de ce livre, à réfléchir au bien-fondé de la question que vous essayez de résoudre à l'aide du machine learning ; à l'utilisation que vous envisagez du modèle que vous êtes en train de développer, et à celles qui pourraient en être faite par d'autres ; et enfin, à la pertinence des données que vous comptez utiliser et aux biais qu'elles pourraient contenir.

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.

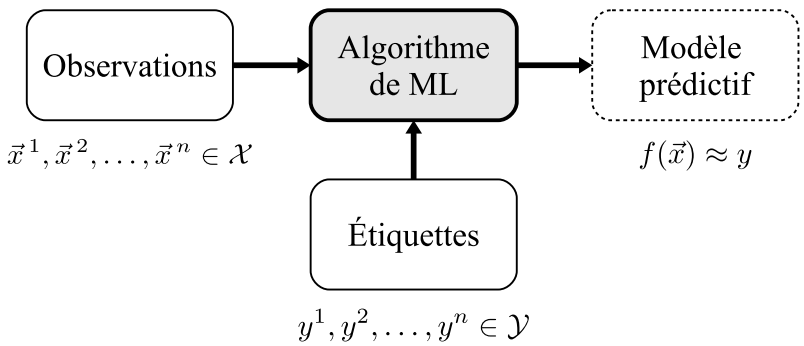

Figure 1.1 – Apprentissage supervisé.

> ***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* $\{\vec{x}^i\}_{i=1,\ldots,n}$ décrites dans un espace $\mathcal{X}$, et leurs *étiquettes* $\{y^i\}_{i=1,\ldots,n}$ 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 : $y^i = \phi(\vec{x}^i) + \epsilon_i$, 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 $(\vec{x}, \phi(\vec{x})) \in \mathcal{X} \times \mathcal{Y}$, $f(\vec{x}) \approx \phi(\vec{x})$.

D'un point de vue probabiliste, on suppose que les données étiquetées $(\vec{x}^i, y^i)$ sont autant de réalisations d'un même couple de variables aléatoires (X, Y), qui vérifie $Y = \phi(X) + \varepsilon$, 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 comment traiter d'autres types de représentations, comme des variables binaires, discrètes, catégoriques, voire des chaînes de caractères ou des graphes.

Classification binaire

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 [1] *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 $\mathcal{Y} = \{1, 2, \ldots, C\}$ 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.

1. Nous utilisons ici la définition bourbakiste de « plusieurs », c'est-à-dire strictement supérieur à deux.

- 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).

Figure 1.2 – Apprentissage non supervisé.

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* $\{\vec{x}^i\}_{i=1,\ldots,n}$ décrites dans un espace $\mathcal{X}$, il s'agit d'apprendre une nouvelle représentation de ces observations, considérée comme plus informative.

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.

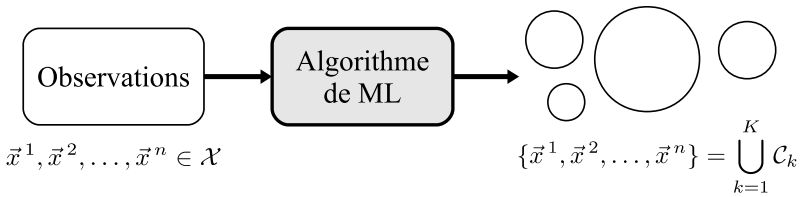

Figure 1.3 – Partitionnement des données, ou clustering.

Définition 1.6 (Partitionnement)

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

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 dans lequel elles sont représentées à l'origine (voir figure 1.4). Cela permet de réduire les temps de calcul et l'espace mémoire nécessaire au stockage les données, mais aussi souvent d'améliorer les performances d'un algorithme d'apprentissage supervisé entraîné par la suite sur ces données.

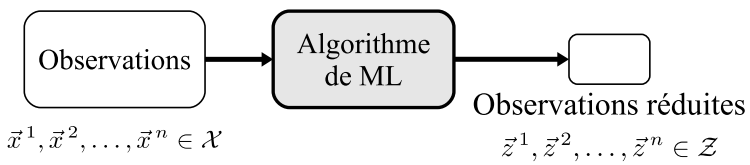

Figure 1.4 – Réduction de dimension.

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* $\{\vec{x}^i\}_{i=1,\ldots,n}$. Les projections $\{\vec{z}^i\}_{i=1,\ldots,n}$ des données sur $\mathcal{Z}$ doivent vérifier certaines propriétés à préciser. Chaque observation est maintenant représentée par sa projection sur $\mathcal{Z}$.

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'annexe 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. Il s'agit d'un sujet plus avancé, que nous ne considérerons pas dans cet ouvrage.

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. Ce sujet dépasse largement le cadre de cet ouvrage.

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 KDNuggets (`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_{ij} 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 $\max(0, a)$;
- $\mathbb{P}(A)$ représente la probabilité de l'événement A ;
- $\mathbb{E}[X]$ représente l'espérance de la variable aléatoire X ;
- $\mathbb{V}[X]$ représente la variance de la variable aléatoire X ;
- $\mathbb{1}$ est la fonction indicatrice

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

- $\langle ., . \rangle$ représente le produit scalaire sur $\mathbb{R}^p$;
- $\langle ., . \rangle_{\mathcal{H}}$ 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.

Pour aller plus loin

- Pour plus de détails sur l'estimation de densité, on consultera le livre de Scott (1992).
- Sur la régression structurée, on pourra se référer à BakIr et al. (2007).
- L'ouvrage de Sutton et Barto (2018) est un bon point de départ pour se plonger dans le sujet de l'apprentissage par renforcement.

Bibliographie

BakIr, 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`.

Benureau, F. (2015). *Self-Exploration of Sensorimotor Spaces in Robots*. Thèse de doctorat, université de Bordeaux.

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.

Scott, D. W. (1992). *Multivariate density estimation*. Wiley, New York.

Sutton, R. S. et Barto, A. G. (2018). *Reinforcement Learning : An Introduction*. MIT Press, Cambridge, MA. http://incompleteideas.net/book/the-book-2nd.html.

Exercices

1.1 Alice veut écrire un programme qui utilise la fréquence des mots « science », « public », « accès », « université », « gouvernement », « financer », « éducation », « budget », « justice » et « loi » pour déterminer si un article traite ou non de politique scientifique. Elle a commencé par annoter un millier d'articles selon leur sujet. Quel genre de problème d'apprentissage automatique doit-elle résoudre ?

1.2 Parmi les problèmes suivants, lesquels se prêtent bien à être traités par le machine learning ?

1. Déterminer l'horaire optimal pour poster un contenu sur une page web.
2. Déterminer le chemin le plus court entre deux nœuds dans un graphe.
3. Prédire le nombre de vélos à mettre en location à chaque station d'un système de location de vélos citadins.
4. Évaluer le prix qu'un tableau de maître pourra atteindre lors d'une vente aux enchères.
5. Débruiter un signal radio.

1.3 Benjamin dispose de 10 000 articles de journaux qu'il souhaite classer par leur thématique. Doit-il utiliser un algorithme supervisé ou non supervisé ?

1.4 Les données de Cécile sont décrites par 10 variables. Elle aimerait cependant les représenter sur un graphique en deux dimensions. Quel type d'algorithme d'apprentissage doit-elle utiliser ?

1.5 David gère un outil qui permet d'organiser les liens HTML qui ont été sauvegardés. Il souhaite suggérer des catégories auxquelles affecter un nouveau lien, en fonction des catégories déjà définies par l'ensemble des utilisateurs du service. Quel type d'algorithme d'apprentissage doit-il utiliser ?

1.6 Elsa veut examiner ses spams pour déterminer s'il existe des sous-types de spams. Quel type d'algorithme d'apprentissage doit-elle utiliser ?

1.7 Tom Mitchell définit le machine learning comme suit : « *Un programme informatique est dit apprendre de l'expérience E pour la tâche T et une mesure de*

performance P si sa performance sur T, comme mesurée par P, s'améliore avec l'expérience E ». Fred écrit un programme qui utilise des données bancaires dans le but de détecter la fraude bancaire. Que sont E, T, et P ?

Solutions

1.1 Apprentissage supervisé (classification binaire).

1.2 1, 3, 4. (2 se résout par des algorithmes de recherche sur graphe, 5 par des algorithmes de traitement du signal).

1.3 Non supervisé.

1.4 Réduction de dimension.

1.5 Apprentissage supervisé (classification multi-classe).

1.6 Apprentissage non supervisé (clustering).

1.7 E = les données bancaires. P = la capacité à détecter correctement une fraude. T = prédire la fraude.

2 APPRENTISSAGE SUPERVISÉ

INTRODUCTION

Dans cet ouvrage, nous nous intéresserons principalement aux problèmes d'apprentissage *supervisé* : il s'agit de développer des algorithmes qui soient capables d'apprendre des modèles *prédictifs*. À partir d'exemples étiquetés, ces modèles seront capables de prédire l'étiquette de nouveaux objets. Le but de ce chapitre est de développer les concepts généraux qui nous permettent de formaliser ce type de problèmes.

OBJECTIFS

- Formaliser un problème comme un problème d'apprentissage supervisé.
- Choisir une fonction de coût.
- Lier la capacité d'un modèle à généraliser avec sa complexité.

2.1 FORMALISATION D'UN PROBLÈME D'APPRENTISSAGE SUPERVISÉ

Un problème d'*apprentissage supervisé* peut être formalisé de la façon suivante : étant données *n observations* $\{\vec{x}^1, \vec{x}^2, \ldots, \vec{x}^n\}$, où chaque observation $\vec{x}^i$ est un élément de l'espace des observations $\mathcal{X}$, et leurs *étiquettes* $\{y^1, y^2, \ldots, y^n\}$, où chaque étiquette y^i appartient à l'espace des étiquettes $\mathcal{Y}$, le but de l'apprentissage supervisé est de trouver une fonction $f : \mathcal{X} \to \mathcal{Y}$ telle que $f(\vec{x}) \approx y$, pour toutes les paires $(\vec{x}, y) \in \mathcal{X} \times \mathcal{Y}$ ayant la même relation que les paires observées. D'un point de vue probabiliste, on suppose que les couples $(\vec{x}^i, y^i)$ sont autant de réalisations d'un couple (X, Y) de variables aléatoires. On souhaite alors que $f(\vec{x}) \approx y$, pour toutes les réalisations $(\vec{x}, y)$ de (X, Y). L'ensemble $\mathcal{D} = \{(\vec{x}^i, y^i)\}_{i=1,\ldots,n}$ forme le *jeu d'apprentissage*.

Nous allons considérer dans cet ouvrage trois cas particuliers pour $\mathcal{Y}$:

- $\mathcal{Y} = \mathbb{R}$: on parle d'un problème de *régression* ;
- $\mathcal{Y} = \{0, 1\}$: on parle d'un problème de *classification binaire*, et les observations dont l'étiquette vaut 0 sont appelées *négatives* tandis que celles dont l'étiquette vaut 1 sont appelées *positives*. Dans certains cas, il sera mathématiquement plus simple d'utiliser $\mathcal{Y} = \{-1, 1\}$;
- $\mathcal{Y} = \{1, 2, \ldots, C\},\ C > 2$: on parle d'un problème de *classification multi-classe*.

Dans de nombreuses situations, on se ramènera au cas où $\mathcal{X} = \mathbb{R}^p$. On dira alors que les observations sont représentées par *p variables*. Dans ce cas, la matrice

$X \in \mathbb{R}^{n \times p}$ telle que $X_{ij} = x_j^i$ soit la j-ème variable de la i-ème observation est appelée *matrice de données* ou *matrice de design*.

Le machine learning étant issu de plusieurs disciplines et champs d'applications, on trouvera plusieurs noms pour les mêmes objets. Ainsi les variables sont aussi appelées *descripteurs*, *attributs*, *prédicteurs*, ou *caractéristiques* (en anglais, *variables, descriptors, attributes, predictors* ou encore *features*). Les *observations* sont aussi appelées *exemples*, *échantillons* ou *points du jeu de donnée* (en anglais, *samples* ou *data points*). Enfin, les étiquettes sont aussi appelées *variables cibles* (en anglais, *labels, targets* ou *outcomes*).

Ces concepts sont illustrés sur la figure 2.1.

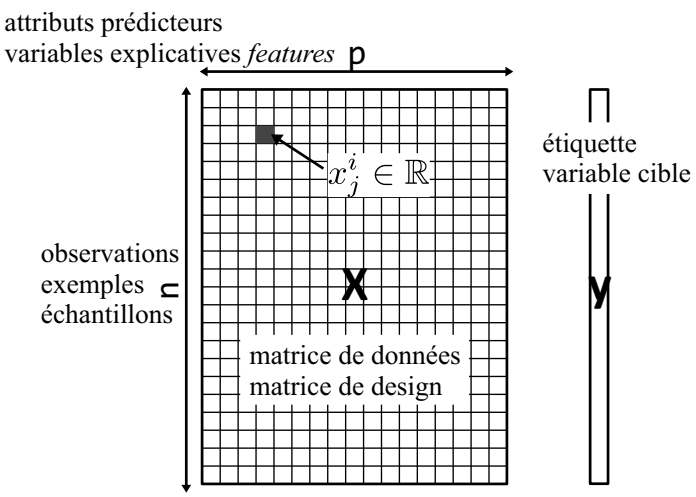

Figure 2.1 – Les données d'un problème d'apprentissage supervisé sont organisées en une matrice de design et un vecteur d'étiquettes. Les observations sont représentées par leurs variables explicatives.

2.1.1 Décision

Dans le cas d'un problème de classification, le modèle prédictif peut prendre directement la forme d'une fonction f à valeurs dans $\{0, 1\}$, ou utiliser une fonction intermédiaire g à valeurs réelles, qui associe à une observation un score d'autant plus élevé qu'elle est susceptible d'être positive. Ce score peut par exemple être la probabilité que cette observation appartienne à la classe positive. On obtient alors f en *seuillant* g ; g est appelée *fonction de décision*.

> ***Définition 2.1 (Fonction de décision)***
>
> Dans le cadre d'un problème de classification binaire, on appelle *fonction de décision*, ou *fonction discriminante*, une fonction $g : \mathcal{X} \mapsto \mathbb{R}$ telle que $f(\vec{x}) = 0$ si et seulement si $g(\vec{x}) \leq 0$ et $f(\vec{x}) = 1$ si et seulement si $g(\vec{x}) > 0$.

Cette définition se généralise dans le cas de la classification *multi-classe* : on a alors C fonctions de décision $g_c : \mathcal{X} \mapsto \mathbb{R}$ telles que $f(\vec{x}) \in \arg\max_{c=1,\dots,C} g_c(\vec{x})$.

Les fonctions de décision permettent de définir les *frontières de décision*, qui séparent les classes les unes des autres.

Définition 2.2 (Frontière de décision)

Dans le cadre d'un problème de classification, on appelle *frontière de décision*, ou *discriminant*, l'ensemble des points de $\mathcal{X}$ où une fonction de décision s'annule. Dans le cas d'un problème binaire, il y a une seule frontière de décision ; dans le cas d'un problème multi-classe à C classes, il y en a C.

2.1.2 Classification multi-classe

Parmi les méthodes d'apprentissage supervisé que présente cet ouvrage, certaines permettent de résoudre directement des problèmes de classification multi-classe. Cependant, tout algorithme de classification binaire peut être utilisé pour résoudre un problème de classification à C classes, par une approche *une-contre-toutes* ou une approche *une-contre-une*.

Définition 2.3 (Une-contre-toutes)

Étant donné un problème de classification multi-classe à C classes, on appelle *une-contre-toutes*, ou *one-versus-all*, l'approche qui consiste à entraîner C classifieurs binaires. Le c-ième de ces classifieurs utilise tous les exemples de la classe c comme exemples positifs, et toutes les autres observations comme exemples négatifs, pour apprendre une fonction de décision g_c. Ainsi, chaque classifieur apprend à distinguer une classe de toutes les autres. L'étiquette de $\vec{x}$ est donnée par celle des fonctions de décision qui retourne le score le plus élevé (ou, si plusieurs d'entre elles retournent le même score maximal, l'une de celles-ci) :

$$f(\vec{x}) \in \arg\max_{c=1,\dots,C} g_c(\vec{x}).$$

Définition 2.4 (Une-contre-une)

Étant donné un problème de classification multi-classe à C classes, on appelle *une-contre-une*, ou *one-versus-one*, l'approche qui consiste à créer $C(C-1)$ classifieurs binaires séparant chacun une classe d'une autre, en ignorant tous les autres exemples. Soit g_{ck} la fonction de décision du classifieur binaire qui sépare la classe c de la classe k. L'étiquette de $\vec{x}$ est déterminée selon :

$$f(\vec{x}) \in \arg\max_{c=1,\dots,C} \left(\sum_{k \neq c} g_{ck}(\vec{x}) \right).$$

Il est difficile de dire laquelle de ces deux approches est la plus performante. En pratique, le choix sera souvent guidé par des considérations de complexité algorithmique : est-il plus efficace d'entraîner C modèles sur n observations, ou $C(C-1)$ modèles sur $\frac{n}{C}$ observations (en supposant les classes équilibrées, autrement dit que