

---

# Chapitre 11

## Les algorithmes d'intelligence artificielle

### 1. Surmonter le paradoxe de Polanyi

Comme nous l'avons vu dans le chapitre L'histoire de l'algorithmique, un algorithme est une suite d'instructions précises permettant d'atteindre un objectif. Les mots-clés sont : précision et objectif. En effet, un ordinateur est un système électromécanique rigoureux qui exécute des millions voire des milliards d'opérations par minute. Il ne fait que ce qui a été explicitement programmé au travers des lignes d'un code du programme. Il ne tolère ni ne gère l'ambiguïté. Si une nouvelle situation non mentionnée dans le programme apparaît, l'ordinateur plante, ou alors ne fait rien, attendant sagement et passivement que le programmeur ajoute les instructions lui permettant de gérer ce cas de figure.

L'ambiguïté apparaît quand les instructions ne sont pas précises ou manquent de clarté, et le langage naturel est bien trop ambigu par moments. N'avez-vous jamais eu du mal à comprendre votre interlocuteur alors qu'aucun mot utilisé ne vous était inconnu ? Savez-vous toujours discerner l'ironie du second degré ? Et même si vous devinez l'intention, êtes-vous certain de placer correctement la ponctuation ? Car elle impacte grandement le sens d'une phrase. Un exemple connu qui illustre son importance est la phrase du comte d'Auteroche lors de la bataille de Fontenoy (1745). La voici sans ponctuation : "Messieurs les Anglais tirez les premiers". Le comte a-t-il voulu dire "Messieurs ! Les Anglais ! Tirez les premiers !" ou plutôt "Messieurs les Anglais, tirez les premiers !" ?

# 166 — Machine Learning et Deep Learning

Des bases à la conception avancée d'algorithmes

C'est d'ailleurs pour réduire l'ambiguïté du langage naturel que des langages de programmation ont été créés. Leur syntaxe rigoureuse et leur implémentation des mécanismes de la logique de programmation rendent le code prévisible, et non pas versatile. Cela permet même à un programme de mettre en place une stratégie de gestion d'erreurs lui stipulant de s'arrêter pour nous proposer des pistes de résolution pour le bogue en cours, voire pour le résoudre automatiquement. Preuve qu'il a "compris" ce que nous "voulions" faire, qu'il a compris le sens de nos instructions. À condition bien sûr que les instructions pour gérer ce cas de figure aient été renseignées dans le programme.

Quel est le problème alors ? Le fait est que parfois décrire une instruction est une tâche plus ardue qu'elle n'y paraît. Par exemple, pourriez-vous rédiger de manière précise la suite d'instructions qui vous permet de reconnaître un visage ? Si vous dites qu'il faut le regarder et vous remémorer les visages que vous connaissez, vous êtes déjà dans l'ambiguïté. En effet, comment faites-vous concrètement et précisément pour vous remémorer un souvenir tel qu'un visage ? Autre exemple : comment faites-vous pour lire un livre ? Savez-vous l'expliquer, juste en utilisant les concepts de variables, conditions, boucles et fonctions ? Peut-être y arriverez vous si ces concepts vous sont familiers. Mais l'expérience montre que la somme d'instructions nécessaire pour cette tâche est incroyablement importante et complexe. Comment pourrait-il en être autrement puisque vous-même ne parvenez pas systématiquement à reconnaître une image précise ou l'ironie dans un texte ? Les concepts d'image, de lecture et d'ironie sont ambigus, tout ce qu'un ordinateur déteste, car il ne sait pas gérer l'ambiguïté.



Cette situation illustre le paradoxe de Karl Polanyi (1886 - 1964), économiste austro-hongrois.

Selon lui, "nous en savons plus que ce que nous pouvons exprimer". Autrement dit, l'être humain réalise des tâches qu'il aurait toutes les peines du monde à expliquer de manière claire et précise. Les humains sont des machines extraordinaires de complexité, dirigées par un cerveau que l'on qualifie comme l'élément le plus complexe de l'univers. Mais pour la plupart de nos tâches, nous fonctionnons en mode automatique. Nous faisons, mais nous ne savons pas exactement comment nous le faisons. La complexité des tâches que nous effectuons est encapsulée dans des mécanismes inconscients qui représentent la partie immergée de l'iceberg. Si nous ne pouvons pas l'expliquer, il ne peut y avoir ni clarté ni précision, et par conséquent, la programmation devient impossible, ou du moins extrêmement compliquée. Comment dépasser ce paradoxe ? Est-il possible de permettre à la machine de "penser" par elle-même ?

# 168 — Machine Learning et Deep Learning

Des bases à la conception avancée d'algorithmes

C'est la question que se pose, en 1950, le mathématicien britannique Alan Turing (1912 - 1954). Bien avant lui, au XVII<sup>e</sup> siècle, des scientifiques rêvaient déjà de fabriquer une machine capable de résoudre des problèmes logiques ou de mathématiques, bref une "machine à penser". Turing, lui, a eu l'idée géniale d'élaborer un test d'intelligence pour les machines, appelé test de Turing. Durant ce test, un humain communique avec deux entités, un autre humain et une machine, avec l'objectif de discerner qui est l'humain et qui est la machine. La machine passe le test avec succès si elle arrive à se rendre indiscernable de l'humain lors de l'échange. Il s'agit donc d'un "jeu d'imitation". Plus le leurre dure longtemps, plus la machine peut être jugée "intelligente". Comment Turing a-t-il eu l'idée d'un test de cette nature ? Il s'est inspiré des travaux du philosophe anglais Thomas Hobbes (1588 - 1679). Ce dernier, en empiriste convaincu, explique dans son ouvrage *Léviathan*, que la pensée et ses conséquences (paroles, actions) ne sont en fait que le résultat d'un calcul. Le terme "calcul" est pris dans un sens très général, les opérations pouvant être de nature mathématique ( $2+2=4$ ) ou simplement logique (syllogisme). La conclusion logique à laquelle on arrive en suivant le raisonnement de Hobbes est qu'une machine à calculer peut être convertie en machine à penser.

C'est dans ce contexte qu'apparaît l'intelligence artificielle, mot inventé par John McCarthy et Marvin Lee Minsky, deux des pionniers de cette discipline visant à surmonter le paradoxe de Polanyi. Leur objectif est basé sur l'hypothèse audacieuse inspirée des travaux de Hobbes selon laquelle "toute activité intellectuelle peut être décrite avec suffisamment de précision pour être simulée par une machine". Les deux ingénieurs définissent l'intelligence artificielle de cette façon : "La construction de programmes informatiques qui s'adonnent à des tâches qui sont, pour l'instant, accomplies de façon plus satisfaisante par des êtres humains, car elles demandent des processus mentaux de haut niveau tels que : l'apprentissage perceptuel, l'organisation de la mémoire et le raisonnement critique." Il s'agit d'une intelligence dans le sens où elle est capable d'autorégulation, d'adaptation, notamment du fait qu'elle gère dans une certaine mesure l'ambiguïté.

Ces avancées sont le résultat de discussions philosophiques entre experts, mais également d'avancées théoriques et pratiques significatives. En effet, en 1955-1956 a été créé le programme Logic Theorist, capable de prouver 38 théorèmes sur 52 des *Principia Mathematica*. En 1957, Frank Rosenblatt invente le perceptron, qui est un algorithme d'apprentissage de classifieurs binaires. Le perceptron peut être considéré comme le type de réseau de neurones le plus simple. Il y a également eu le robot ELIZA, développé par Joseph Weizenbaum entre 1964 et 1966, et dont le but est de simuler un psychothérapeute appliquant les méthodes de Carl Rogers. ELIZA reformule en questions les affirmations du patient et les lui pose.

## 2. Quelques applications de l'intelligence artificielle

Des applications qui auraient été pour ainsi dire impossibles avec des algorithmes utilisant les mécanismes classiques deviennent possibles grâce à l'intelligence artificielle. Cela permet non seulement de réaliser de nouvelles tâches, jusque-là l'apanage des humains, mais aussi de les surpasser dans des domaines toujours plus nombreux. Cela est dû notamment aux quatre éléments suivants dans lesquels les machines nous surpassent :

- Puissance : comparés aux humains, les ordinateurs sont plus résistants et endurants. De plus, ils possèdent une puissance de calcul supérieure bien plus élevée.
- Fiabilité : ils stockent l'information de manière fiable et durable, et cela en très grande quantité au travers de dispositifs toujours plus petits.
- Rapidité : ils sont extrêmement rapides, effectuant des millions de calculs à la seconde.
- Connexion : grâce au réseau Internet qui parcourt le monde, notamment au travers de câbles continentaux, ils peuvent entrer en contact avec des milliards d'autres machines intelligentes et autonomes pour s'envoyer des informations qui circulent à la vitesse de la lumière.

# 170 — Machine Learning et Deep Learning

Des bases à la conception avancée d'algorithmes

Voici une liste non exhaustive des prouesses de l'intelligence artificielle :

- reconnaître des images
- conduire des voitures
- deviner le style d'une musique
- dessiner une pochette d'album
- jouer à des jeux vidéo (échec, go, Mario Kart...)
- composer de la musique
- écrire un article de journal
- acheter et vendre des actions en bourse
- prendre la place de l'arbitre dans une compétition de gymnastique
- diagnostiquer des maladies
- traduire un message oral en texte après avoir déterminé la langue parlée
- etc.

Détaillons quelques exemples issus de deux secteurs distincts, tout en gardant à l'esprit que le champ d'application de l'intelligence artificielle est pour ainsi dire illimité :

- Industrie : ce secteur utilise déjà massivement des produits issus de l'intelligence artificielle. On pense par exemple aux ateliers de fabrication automatisés dans lesquels des robots industriels s'affairent à la fabrication d'objets divers et variés. Un domaine très intéressant concerne la maintenance prédictive, qui consiste à anticiper les pannes d'équipement en fonction du cycle de vie et de l'usage des composants. Il s'agit non seulement d'une problématique de sécurité, mais également d'une question financière. En effet, plutôt que de changer un composant ou un ensemble de composants à date fixe, ce qui peut mener à des actions de maintenance excessives susceptibles de se révéler dangereuses, on personnalise les actions de maintenance pour maximiser la sécurité et donc minimiser les coûts, l'intelligence artificielle permettant d'obtenir une probabilité d'apparition d'une défaillance en fonction des données récoltées par des capteurs et analysées en temps réel. Ces capteurs peuvent alimenter le modèle d'intelligence artificielle avec des données issues de mesures acoustiques, thermographiques ou encore d'analyses vibratoires.

- Commerce : pour vendre plus et mieux, les entreprises veulent analyser et anticiper les comportements clients, dans le but de les regrouper par similarité et de leur proposer ainsi les produits les plus adaptés ou des produits de substitution, et cela au meilleur moment. Cela a également un effet sur la gestion des stocks, car savoir ce que les clients veulent vraiment permet de prévoir la demande d'un produit particulier ou d'un type de produit. L'intelligence artificielle permet d'analyser un ensemble de données (météo, emplacement du magasin, type d'achat, période de l'année...) pour atteindre ces objectifs, et d'autres encore, comme prévoir l'impact d'une campagne publicitaire ou promotionnelle sur le chiffre d'affaires. Parmi les entreprises qui font appel à l'intelligence artificielle, on peut citer Amazon, qui a mis au point un système basé sur les réseaux de neurones (deep learning) nommé DSSTNE (*Deep Scalable Sparse Tensor Network Engine*), dont l'objectif est de recommander des produits en fonction de l'historique des clients (historique qui comprend les produits achetés ou simplement les pages consultées). Il y a également Netflix, qui utilise massivement un système de recommandation de films très perfectionné. Non seulement ce système recommande des films en utilisant la méthode du filtrage collaboratif, mais il analyse également les recherches des utilisateurs pour orienter la stratégie Netflix de création de films, l'entreprise étant passée du statut de simple distributeur de vidéos à producteur à succès.

## Chapitre 4-4

# L'algorithme k-means

### 1. Objectif du chapitre

Les chapitres précédents ont abordé des exemples de deux types d'algorithmes de Machine Learning : les algorithmes de régression et de classification. Ce chapitre porte sur l'algorithme k-means, appelé l'algorithme des k-moyennes en français, qui est un algorithme simple à comprendre et qui fait partie des algorithmes de clustering les plus connus et les plus utilisés.

L'algorithme k-means a été introduit par J. McQueen en 1967. C'est un algorithme non supervisé qui permet de répartir un ensemble de  $n$  observations en  $k$  clusters. L'objectif après l'application de l'algorithme k-means sur un jeu de données est que chaque cluster contienne des observations homogènes et que deux observations de deux clusters différents soient hétérogènes.

Les domaines d'application de l'algorithme k-means sont nombreux. Par exemple, il est très utilisé pour la segmentation des clients à des fins de marketing, ou encore pour l'isolation des motifs dans les images, car justement les images présentent souvent des régions homogènes, notamment en matière d'intensité lumineuse.

De manière générale, le succès de l'algorithme k-means et de ses versions réside dans sa simplicité et sa capacité à traiter des données de grande taille.



# 448 — Le Machine Learning avec Python

De la théorie à la pratique

À la fin de ce chapitre, le lecteur aura abordé :

Le fonctionnement de k-means via des illustrations.

- Les étapes principales de l'algorithme k-means classique.
- L'application de l'algorithme k-means avec Scikit-learn.
- L'impact des valeurs extrêmes sur les performances de l'algorithme k-means.
- La recherche de la valeur optimale du paramètre  $K$  de l'algorithme k-means.
- Les avantages et les inconvénients ainsi que les variantes de l'algorithme k-means.

## 2. k-means du point de vue géométrique

Comme précisé au début de ce chapitre, l'algorithme k-means est très intuitif et simple à comprendre. Avant d'entrer dans les détails, il faut noter que k-means, comme tous les algorithmes de clustering, ne nécessite pas l'étiquetage des données, car c'est une procédure non supervisée.

De façon informelle, étant donné  $n$  observations à répartir sur  $k$  clusters, k-means choisit initialement, de manière aléatoire,  $k$  observations parmi les  $n$  observations, comme étant les centres des  $k$  clusters recherchés. Chacune des  $n$  observations sera associée au cluster dont le centre est le plus proche parmi les  $k$  centres choisis initialement. Une fois que toutes les observations sont associées à leurs clusters respectifs, le centre de chaque cluster est recalculé en fonction des observations qu'il contient. Puis, de nouveau, chacune des observations est associée au cluster dont le centre est le plus proche de cette observation par rapport à tous les centres des autres clusters. Ces opérations de recalcul des centres des clusters puis d'association des observations aux clusters les plus proches sont répétées jusqu'à ce qu'un critère d'arrêt soit atteint.

L'algorithme k-means utilise une fonction pour calculer les distances entre les observations et les centres des clusters. Ce calcul des distances peut être basé sur la distance euclidienne, la distance de Manhattan ou toute autre fonction permettant de mesurer la dissimilarité entre les observations.

Pour mieux comprendre cet algorithme de clustering, cette section déroule l'algorithme k-means sur un exemple simple. Soit six observations a, b, c, d, e et f à répartir sur deux clusters  $C_1$  et  $C_2$  ; supposons que la distance utilisée est la distance euclidienne classique. Ces six observations sont définies dans un espace à deux dimensions et leurs coordonnées sont indiquées dans le tableau suivant :

Axes	a	b	c	d	e	f
x	2	4	2	4	10	10
y	4	4	2	2	2	4

Figure 10-1 : un simple jeu de données avec leurs coordonnées en deux dimensions

#### ■ Remarque

Pour rappel, la distance euclidienne entre deux observations  $A=(x_A, y_A)$  et  $B=(x_B, y_B)$  est calculée grâce à la formule :

$$\sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

Avant de faire un traitement quelconque sur les données, il est toujours intéressant de les visualiser lorsque c'est possible. Dans cet exemple, les données sont définies dans un espace à deux dimensions, donc elles peuvent être facilement visualisées sur deux axes comme dans la figure 10-2 ci-dessous.

#### ■ Remarque

Même lorsque les données sont définies dans un espace à grande dimension, supérieur à deux ou à trois dimensions, il existe des méthodes qui permettent de les visualiser en deux ou trois dimensions, avec une perte d'informations qu'on espère minimale. Ces méthodes sont appelées les méthodes de réduction de domaines. Le chapitre Analyse en composantes principales présente l'analyse du même nom, qui est l'une des méthodes de réduction de domaines les plus connues et qui permet d'avoir une vue en deux dimensions des données définies initialement dans un espace à grande dimension.

# 450 — Le Machine Learning avec Python

De la théorie à la pratique

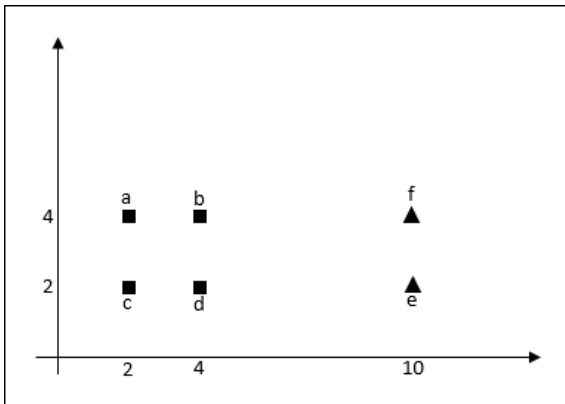


Figure 10-2 : représentation graphique en deux dimensions des données

Ce graphique montre clairement que les observations a, b, c et d, représentées par des carrés, sont très proches entre elles au sens de la distance euclidienne, par rapport aux deux observations e et f. Également, les deux dernières observations, représentées par des triangles, sont très proches entre elles.

Pour cet exemple, en se basant donc sur la distance euclidienne, un algorithme de clustering efficace proposerait certainement de répartir ces six observations dans les deux clusters  $C_1$  et  $C_2$  comme dans la figure 10-3 ci-dessous.

En effet, avec la distance euclidienne, cette répartition est optimale. La section suivante définit de façon plus formelle la notion de solution optimale pour un algorithme k-means et pour un nombre de clusters fixe.

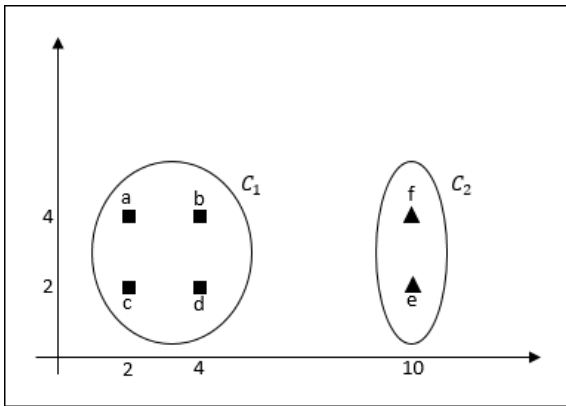


Figure 10-3 : répartition des six points dans les deux classes  $C_1$  et  $C_2$

En suivant les étapes classiques de l'algorithme k-means, le résultat optimal de la figure 10-3 peut être obtenu comme suit :

1. L'algorithme k-means commence initialement par sélectionner de façon aléatoire deux observations parmi les six observations disponibles. Les deux observations ainsi sélectionnées vont être considérées comme les centres des deux clusters recherchés  $C_1$  et  $C_2$ . Ici, nous supposons que l'algorithme k-means recherche un nombre de clusters qui est égal à 2.

Dans cet exemple, supposons que les deux observations a et d sont sélectionnées aléatoirement. Ces deux observations vont être considérées comme les centres respectifs des clusters  $C_1$  et  $C_2$ . L'algorithme k-means calcule les distances entre chacune des six observations avec les centres a et d. Les résultats sont reportés dans le tableau ci-dessous :

Les centres des clusters	a	b	c	d	e	f
Centre de $C_1$ =a=(2,4)	0	2	2	2.8284	8.2462	8
Centre de $C_2$ =d=(4,2)	2.8284	2	2	0	6	6.3245

Figure 10-4 : distances entre les observations et les centres de  $C_1$  et  $C_2$

2. Une fois que k-means dispose de toutes les distances entre toutes les observations et les deux centres a et d, il procède à l'association entre les observations et les clusters. Par exemple, l'observation e va être associée au cluster  $C_2$ , puisqu'elle est plus proche du centre de  $C_2$  que du centre de  $C_1$ . À la suite de cette étape, les deux clusters  $C_1$  et  $C_2$  vont être constitués comme suit  $C_1 = \{a, b, c\}$  et  $C_2 = \{d, e, f\}$

Lorsqu'une observation est à la même distance des clusters  $C_1$  et  $C_2$ , alors elle est affectée à l'un de ces deux clusters de manière aléatoire. Dans notre exemple nous avons affecté de manière arbitraire les deux observations b et c au cluster  $C_1$ .

La figure suivante présente graphiquement les deux clusters  $C_1$  et  $C_2$  obtenus à la suite de cette étape :

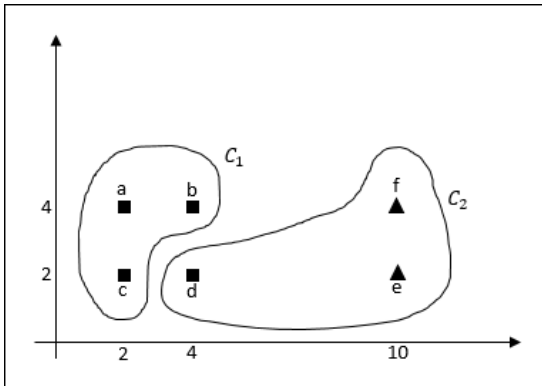


Figure 10-5 : affectation des observations aux clusters  $C_1$  et  $C_2$  après la première itération

3. Lors de la première étape, l'algorithme k-means a choisi de façon aléatoire a et d en tant que centres des deux clusters recherchés. À l'étape 2, les deux clusters  $C_1$  et  $C_2$  ont été concrètement construits.

À cette étape, l'algorithme k-means calcule le centre du cluster  $C_1$  en utilisant les observations a, b et c et calcule le centre du cluster  $C_2$  en utilisant les observations d, e et f.

Ainsi : le centre de  $C_1 = \left(\frac{1}{3}(2 + 2 + 4), \frac{1}{3}(4 + 4 + 2)\right) = (2.66, 3.33)$

le centre de  $C_2 = \left(\frac{1}{3}(4 + 10 + 10), \frac{1}{3}(2 + 2 + 4)\right) = (8, 2.66)$

Les centres des deux clusters  $C_1$  et  $C_2$  sont représentés avec le symbole étoile dans la figure 10-6 ci-dessous :

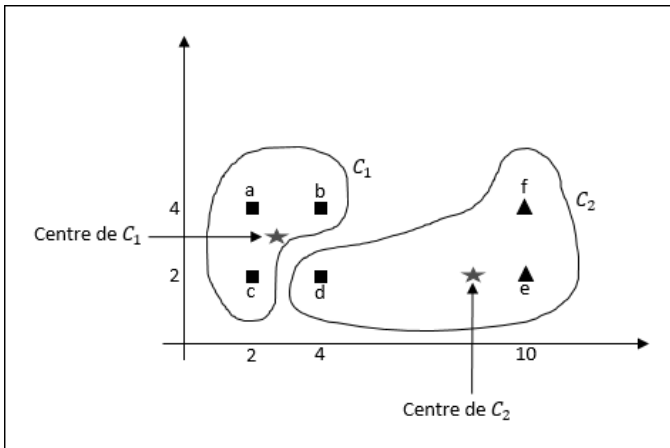


Figure 10-6 : les centres des clusters  $C_1$  et  $C_2$

- À cette étape, les six observations sont réparties de nouveau sur les deux clusters  $C_1$  et  $C_2$  en se basant sur les nouveaux centres calculés à l'étape précédente. Les distances entre les observations et les centres de  $C_1$  et  $C_2$  sont reportées dans le tableau suivant :

Les centres des clusters	a	b	c	d	e	f
Centre de $C_1 = (2.66, 3.33)$	0.9404	1.4981	1.4847	1.8879	7.4595	7.3505
Centre de $C_2 = (8, 2.66)$	6.1478	4.2184	6.0361	4.0540	2.1060	2.4074

Figure 10-7 : distances entre les observations et les centres de  $C_1$  et  $C_2$