Intelligence artificielle

Apprentissage automatique

Apprentissage automatique

Un **programme informatique** a pour objectif de résoudre un problème dont on connaît la solution. Par exemple, le logiciel Pronote sait calculer la moyenne générale d'un élève à partir de toutes les notes entrées par les professeurs.

Pour les problèmes que l'on ne sait pas résoudre, comme la reconnaissance automatique des chiffres écrits à la main à partir d'une image scannée, on ne peut pas écrire de programme classique. Mais il est facile de disposer d'une base de données regroupant de nombreuses instances du problème considéré.

L'apprentissage automatique consiste alors à écrire un algorithme qui, à partir de données et d'expériences passées, va apprendre à **résoudre au mieux** un problème considéré.

Mode d'apprentissage

On distingue les algorithmes **supervisés** des algorithmes **non supervisés**.

Prenons un exemple d'images représentant des chats ou des chiens :

— s'il s'agit d'un algorithme supervisé, c'est nous, humains, qui proposons les critères de classement (« chat » ou « chien »). Le nombre de classes est connu ;

— l'algorithme non supervisé doit trouver lui-même les critères pour organiser les données en groupes : on parle alors de **clustering**.

Type de problème

On distingue les algorithmes de **régression** de ceux de **classification**.

L'exemple précédent est un algorithme de classification : la nouvelle image correspond à un chien ou un chat, sans possibilité intermédiaire. On parle de *valeur discrète*.

Un algorithme de régression permet d'obtenir un *nombre réel*. Par exemple, à partir d'un échantillon contenant les données des habitations d'une ville (quartier, superficie, prix, etc.), l'algorithme sera en mesure de proposer un prix pour une nouvelle habitation.

Répartition des données

Entraîner un modèle, c'est ajuster pas à pas ses paramètres pour que ses prédictions se trompent le moins possible sur les exemples dont on dispose. À la fin de l'entraînement, on garde le **modèle** qui a donné les meilleurs résultats.

On ne peut pas juger ce modèle sur les exemples qui ont servi à l'entraîner : il les a, en quelque sorte, appris par cœur, comme un élève qui réussit les exercices dont il a lu le corrigé. On partage donc les données en deux groupes dès le départ.

Les données d'entraînement

C'est la plus grande part des données, généralement **80 %**. Le modèle les utilise pour ajuster ses paramètres.

Les données de test

Ce sont les **20 %** restants, jamais montrés au modèle pendant l'entraînement. On compare ses prédictions sur ces données aux vraies valeurs : le taux d'erreur obtenu indique ce que le modèle fera sur des données nouvelles. On ne modifie plus le modèle après avoir vu ce résultat.

Algorithmes d'apprentissage automatique

Supervisés

Parmi les algorithmes supervisés les plus courants : régression linéaire, régression logistique, classification naïve bayésienne, forêts d'arbres de décision, machines à vecteurs de support et k plus proches voisins.

Non supervisés

Parmi les algorithmes non supervisés, on trouve les méthodes de réduction de dimensionnalité et l'algorithme des k-moyennes.

Algorithme des k-moyennes (*k-means*) : on classe les données par groupes (*clusters*) de similarités. On utilise souvent la notion de distance pour faire ces regroupements.

Exemples d'applications

L'apprentissage automatique est utilisé dans de très nombreux domaines : médecine (soutien au diagnostic), finance (trading algorithmique), police (prévention des crimes), justice (aide à la décision), médias (analyse de contenu multimédia), musique et images (génération à partir d'une description en langage naturel), services client (assistants conversationnels), jeux vidéo (navigation des personnages non joueurs), transports (voitures autonomes), éducation (parcours personnalisés), cybersécurité (prévention des attaques), plateformes de streaming (recommandations personnalisées), GPS (anticipation des embouteillages) et traduction (analyse sémantique).

Les modèles de langage

Depuis 2022, les applications les plus visibles de l'apprentissage automatique sont les **modèles de langage** (*large language models*, LLM) tels que ChatGPT, Claude, Gemini ou Mistral. Un modèle de langage est un réseau de neurones entraîné sur des milliards de textes à une seule tâche : prédire le mot (plus précisément le *jeton*) le plus probable après ceux qu'il a déjà lus. Répétée mot après mot, cette prédiction produit des réponses, des traductions, des résumés ou du code.

Trois points à retenir :

— un modèle de langage ne consulte pas une base de faits : il **génère** une suite plausible, ce qui explique qu'il puisse affirmer avec assurance des choses fausses (on parle d'*hallucinations*) ;

— ses connaissances s'arrêtent à la date de ses données d'entraînement, sauf si on lui fournit des documents ou un accès à la recherche en ligne ;

— son entraînement demande des quantités considérables de données, de calcul et d'énergie, et pose des questions de droit d'auteur, de biais et de confidentialité.

On parle plus largement d'**IA générative** pour les modèles qui produisent du texte, des images, du son ou de la vidéo.

Algorithme des k plus proches voisins

L'**algorithme des k plus proches voisins** est un **algorithme d'apprentissage supervisé**, souvent noté k-NN pour *k-nearest neighbors*.

Lorsqu'on exécute l'algorithme sur une nouvelle donnée, la méthode consiste à prendre en compte les k échantillons d'apprentissage les plus proches pour calculer le résultat. La classe majoritairement représentée est choisie pour classer la nouvelle donnée.

Reprenons l'exemple des images de chiens ou de chats :

— choisissons $k = 5$: l'algorithme ne prend en compte que les cinq images dont les caractéristiques sont les plus proches de l'image à tester ;

— supposons que, parmi ces cinq images, trois sont celles de chats et deux celles de chiens : l'algorithme considère que la nouvelle image est celle d'un chat ;

— on dit parfois que l'on fait « voter » les plus proches voisins.

Dans cet algorithme, la notion de **distance entre échantillons** est primordiale et doit être définie en fonction des données et du fonctionnement attendu. Le choix de k peut changer le résultat : si, parmi les trois images les plus proches, deux sont celles de chiens et une celle d'un chat, l'algorithme avec $k = 3$ classe la nouvelle image en tant que chien, alors qu'avec $k = 5$ il la classait en tant que chat.

Une grande part de l'apprentissage automatique consiste à entraîner l'algorithme afin de minimiser les erreurs de prédiction. Pour l'algorithme k-NN, cela consiste, entre autres, à :

— ne choisir que les paramètres pertinents des données (est-il judicieux de prendre en compte le paysage de fond dans les photos ?) ;

— choisir le meilleur nombre k de voisins à faire voter (est-il judicieux de choisir un k pair ? trop petit ? trop grand ?) ;

— choisir la meilleure formule pour le calcul de la distance (euclidienne, de Manhattan, de Minkowski, de Tchebychev, etc.).