Comment une IA prend-elle une décision ?

Les arbres de décision

Mise en situation. Chaque jour, votre messagerie range automatiquement certains messages dans le dossier « indésirables » : ce sont les **spams** (publicités, arnaques…). Comment le logiciel décide-t-il ?
Une des méthodes les plus simples d'intelligence artificielle consiste à poser au message une suite de **questions fermées** (réponse oui ou non), comme dans le jeu « Qui est-ce ? ». Cette suite de questions forme un **arbre de décision**. Son grand avantage : contrairement à un réseau de neurones, on peut **lire** l'arbre et **expliquer** chaque décision.
Dans cette activité, vous allez lire des arbres de décision, en construire un à partir d'exemples, puis mesurer s'il se trompe.

1. Le jeu des questions

▶ À vous de jouer !

Une personne pense à un nombre entier entre 1 et 16. Pour le trouver, on ne peut poser que des questions auxquelles elle répond par oui ou par non.

1. On pose la question « Est-ce 7 ? ». Combien de nombres restent possibles si la réponse est non ?
2. On pose plutôt la question « Le nombre est-il plus grand que 8 ? ». Combien de nombres restent possibles si la réponse est oui ? Et si elle est non ?
3. Quelle est la meilleure des deux questions ? Pourquoi ?
4. En posant à chaque fois une question qui partage les nombres possibles en deux groupes égaux, combien de questions faut-il pour être sûr de trouver le nombre ?
5. Combien de questions faut-il pour un nombre entier entre 1 et 1 000 ? On pourra calculer 2^9 et 2^{10}.

Coup de pouce

Pour la question 4, on part de 16 nombres possibles, et chaque bonne question divise ce nombre par 2 : $16 \to 8 \to \dots$

Remarque

Une « bonne » question est une question qui **sépare** le mieux les possibilités. C'est exactement ce que cherche un algorithme qui fabrique un arbre de décision.

2. Lire un arbre de décision

Définition

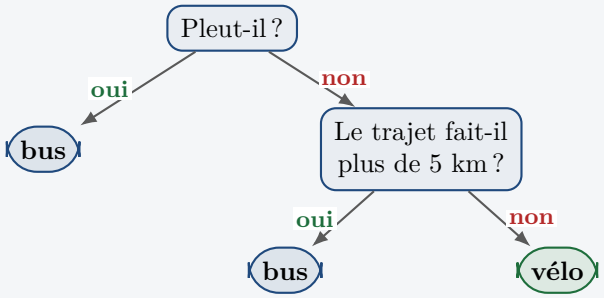
Un **arbre de décision** est un schéma qui pose des questions les unes après les autres :

- chaque **nœud** (rectangle bleu) contient une question ; la première question s'appelle la **racine** ;
- chaque **branche** correspond à une réponse (oui ou non) ;
- chaque **feuille** (en bout de branche) donne la **décision**.

Pour utiliser l'arbre, on part de la racine et on suit les branches qui correspondent aux réponses, jusqu'à une feuille.

Exemple

Pour aller au lycée, Inès choisit entre le vélo et le bus avec l'arbre suivant :



▶ À vous de jouer !

1. Combien cet arbre a-t-il de questions ? de feuilles ? Combien de questions Inès pose-t-elle au maximum ?
2. Quel moyen de transport Inès choisit-elle dans chacun des cas suivants ?

Jour	lundi	mardi	mercredi	jeudi
Pluie	non	oui	non	non
Longueur du trajet	3 km	2 km	8 km	5 km

3. Mardi, le trajet ne fait que 2 km : pourquoi Inès ne se pose-t-elle même pas la question de la distance ?

3. Apprendre un arbre à partir d'exemples

Pour fabriquer un filtre anti-spam, on dispose de 12 messages déjà classés par un humain. Pour chacun, on répond à trois questions : **C** : l'expéditeur est-il dans mes contacts ? **L** : le message contient-il un lien ? **G** : le message contient-il le mot « gratuit » ou un mot de la famille de « gagner » ?

n°	Expéditeur	Message	C	L	G	Classement
1	Mamie	Bisous, à dimanche !	oui	non	non	non-spam
2	Léa	Regarde cette vidéo gratuite : lien	oui	oui	oui	non-spam
3	Hugo	J'ai gagné mon match de foot !	oui	non	oui	non-spam
4	Tom (compte piraté)	Tu as gagné un téléphone, clique ici : lien	oui	oui	oui	spam
5	Maman	Concert gratuit samedi, on y va ?	oui	non	oui	non-spam
6	inconnu	Félicitations, vous avez gagné 1 000 € ! lien	non	oui	oui	spam
7	inconnu	Colis gratuit en attente, suivez le lien : lien	non	oui	oui	spam
8	inconnu	Vous êtes le grand gagnant ! Répondez OUI.	non	non	oui	spam
9	club de foot	Confirme ton inscription : lien	non	oui	non	non-spam
10	inconnu	Jeu gratuit, téléchargez-le vite : lien	non	oui	oui	spam
11	inconnu	Cadeau gratuit pour vous, appelez ce numéro	non	non	oui	spam
12	lycée	Atelier gratuit d'aide aux devoirs mardi	non	non	oui	non-spam

Remarque

Les 12 messages ont été classés à l'avance par un humain : ce sont des données **étiquetées**. Apprendre à partir d'exemples étiquetés s'appelle l'**apprentissage supervisé** (comme pour les k plus proches voisins ou les réseaux de neurones). Avec les k-moyennes, au contraire, aucune étiquette n'était donnée : c'était de l'apprentissage **non supervisé**.

Définition – construire un arbre

Pour choisir la question à placer dans un nœud, l'algorithme essaie **chaque question** :

1. il partage les messages en deux groupes (réponse oui / réponse non) ;
2. dans chaque groupe, il prend la décision **majoritaire** (spam ou non-spam) ;
3. il compte les **erreurs** : les messages du groupe qui ne sont pas de la catégorie majoritaire.

Il garde la question qui fait le **moins d'erreurs**. Il recommence ensuite dans chaque branche, tant qu'une nouvelle question diminue le nombre d'erreurs ; sinon, la branche se termine par une feuille.

Exemple

Sans poser aucune question, on a 6 spams et 6 non-spams : quelle que soit la décision, on se trompe 6 fois.

Rappel – **C** : l'expéditeur est-il dans mes contacts ? **L** : le message contient-il un lien ? **G** : contient-il « gratuit » ou un mot de la famille de « gagner » ?

▶ À vous de jouer !

1. Compléter le tableau ci-dessous pour chacune des trois questions posées à la racine. Quelle question l'algorithme place-t-il à la racine ?

Question	Groupe	Spams	Non-spams	Erreurs
C	oui			
	non			
L	oui			
	non			
G	oui			
	non			

2. On s'intéresse aux 7 messages d'expéditeurs inconnus (C = non). Combien d'erreurs fait-on sans poser de nouvelle question ? Avec la question L ? Avec la question G ? Quelle question place-t-on dans cette branche ?
3. On s'intéresse aux 5 messages d'expéditeurs connus (C = oui). Une nouvelle question (L ou G) diminue-t-elle le nombre d'erreurs ? Que place-t-on au bout de cette branche ?
4. Dessiner l'arbre obtenu. Combien des 12 messages sont bien classés ? En déduire le pourcentage de réussite sur ces messages, arrondi à l'unité.

Coup de pouce

Pour la question C : le groupe « oui » contient les messages 1 à 5, le groupe « non » contient les messages 6 à 12. Dans chaque groupe, on compte les spams et les non-spams ; les erreurs sont le plus petit des deux nombres.

Remarque

Les vrais algorithmes (appelés CART ou ID3) mesurent la qualité d'une question avec des formules un peu plus fines que le nombre d'erreurs, mais l'idée est la même : choisir, à chaque nœud, la question qui **sépare le mieux** les catégories.

4. L'arbre fonctionne-t-il sur de nouveaux messages ?

Un arbre réussit forcément bien sur les messages qui ont servi à le construire (le **jeu d'entraînement**). Pour savoir s'il est vraiment utile, on le teste sur des messages qu'il n'a jamais vus (le **jeu de test**).

Rappel – **C** : l'expéditeur est-il dans mes contacts ? **L** : le message contient-il un lien ? **G** : contient-il « gratuit » ou un mot de la famille de « gagner » ?

n°	Expéditeur	Message	C	L	G	Vrai classement
T1	inconnu	Votre compte est bloqué, connectez-vous : lien	non	oui	non	spam
T2	Papa	On mange à 19 h	oui	non	non	non-spam
T3	inconnu	Gagnez un voyage gratuit : lien	non	oui	oui	spam
T4	professeure	Rappel : sortie au musée vendredi	non	non	non	non-spam
T5	Clara	J'ai gagné ma place pour le concert !	oui	non	oui	non-spam
T6	inconnu	Smartphone gratuit, répondez vite	non	non	oui	spam

▶ À vous de jouer !

1. À l'aide de l'arbre construit dans la partie 3, classer chacun des 6 messages de test.
2. Combien de messages sont bien classés ? Calculer le pourcentage de réussite sur le jeu de test, arrondi à l'unité.
3. Quel message est mal classé ? Ce type de message est-il dangereux ? Proposer une question supplémentaire qui permettrait de le repérer.
4. Pourquoi ne faut-il pas tester l'arbre sur les messages qui ont servi à le construire ?
5. Pourquoi dit-on que cet arbre a été obtenu par **apprentissage supervisé** ? Qui a joué le rôle du « superviseur » ?

5. Programmer l'arbre en Python

Exemple

Un arbre de décision se traduit directement par des instructions conditionnelles `if` / `elif` / `else`. Les réponses oui et non sont représentées par les booléens `True` et `False`.

```
def est_spam(connu, gratuit):
    if connu:
        return False
    elif gratuit:
        return True
    else:
        return False
```

▶ À vous de jouer !

1. Que renvoient `est_spam(True, True)` et `est_spam(False, True)` ? À quels messages du jeu d'entraînement ces appels correspondent-ils ?
2. Écrire, sur le même modèle, une fonction `transport(pluie, distance)` qui renvoie `"bus"` ou `"vélo"` en suivant l'arbre d'Inès (partie 2).
3. On stocke les réponses et les vrais classements des 6 messages de test dans des listes. Compléter le programme qui compte les messages bien classés :

```
connus   = [False, True, False, False, True, False]
gratuits = [False, False, True, False, True, True]
spams    = [True, False, True, False, False, True]

bons = 0
for i in range(len(spams)):
    if est_spam(..., ...) == spams[i]:
        bons = ...
print(bons, "sur", len(spams))
```

Qu'affiche ce programme ?

Coup de pouce

Dans la fonction `transport`, la deuxième question s'écrit `distance > 5`.

6. Pour aller plus loin : expliquer une décision

Remarque

Un arbre de décision est **explicable** : pour chaque message, on peut dire pourquoi il a été classé spam (« expéditeur inconnu et présence du mot gratuit »). Ce n'est pas le cas d'un réseau de neurones, qui combine des milliers de nombres. C'est important : en France, depuis la loi pour une République numérique (2016), une administration qui prend une décision individuelle à l'aide d'un algorithme doit pouvoir en expliquer les règles à la personne concernée.
En pratique, on utilise souvent des **forêts aléatoires** : des centaines d'arbres, construits sur des parties différentes des données, qui **votent** pour prendre la décision (comme le vote des k plus proches voisins).

Rappel – **C** : l'expéditeur est-il dans mes contacts ? **L** : le message contient-il un lien ? **G** : contient-il « gratuit » ou un mot de la famille de « gagner » ?

★ Question étoilée

Un élève veut un arbre qui ne fasse **aucune erreur** sur les 12 messages d'entraînement, en utilisant les trois questions C, L et G.

1. Combien y a-t-il de façons différentes de répondre aux trois questions C, L et G ?
2. Comparer les réponses des messages 2 et 4, puis celles des messages 8, 11 et 12. Expliquer pourquoi l'arbre parfait est impossible.
3. L'élève ajoute la question « Le message vient-il du lycée ? ». Son arbre ne fait plus qu'une erreur sur l'entraînement. Cette question sera-t-elle utile pour classer de nouveaux messages ? Comment appelle-t-on ce défaut ?

À retenir

- Un **arbre de décision** pose des questions fermées les unes après les autres ; chaque feuille donne une décision. Il se traduit directement par des `if` / `elif` / `else`.
- C'est de l'**apprentissage supervisé** : l'arbre est construit à partir d'exemples **étiquetés** par un humain.
- Pour le construire, l'algorithme choisit à chaque nœud la question qui **sépare le mieux** les exemples (ici, celle qui fait le moins d'erreurs).
- On l'évalue sur un **jeu de test** : des exemples qu'il n'a jamais vus. Un arbre trop précis sur l'entraînement fait du **sur-apprentissage**.
- Un arbre est **explicable** : on peut justifier chacune de ses décisions.