

# Briève introduction à l'apprentissage machine

Nicolas Hurtubise  
Vincent Antaki

\*Ou introduction aux modèles d'apprentissages non-paramétrés

*hurtubin@iro.umontreal.ca*  
*antakivi@iro.umontreal.ca*

# L'apprentissage machine ?

Selon **Sébastien Gambs**

*L'apprentissage machine étudie les techniques permettant de donner à la machine la capacité d'apprendre à partir d'expériences passées*

---

## Quel rapport avec l'UdeM ?

- Pionnière de la technique des réseaux profonds (le modèle trendy en ce moment) avec l'Université de Toronto et l'Université de New York.
  - Un pas mal gros laboratoire d'apprentissage-machine.
-

# Grosso-modo c'est quoi ?

*Champ d'étude de l'intelligence artificielle visant à apprendre à partir d'exemples les paramètres d'un modèle en vue d'accomplir une tâche.*

# Un modèle ?

- Le modèle est la partie la plus importante de tout algorithme d'apprentissage. Un modèle définit une fonction de décision et, du coup, les paramètres à apprendre.
- Ex. : Une ligne peut servir à classer un ensemble en 2 sections.  
 $f(x) : ax + b$   
Par exemple, nous avons la position et l'équipe des joueurs sur un terrain de ballon-chasseur. Nous cherchons à estimer la position de la ligne du milieu du terrain en fonction des joueurs.

Mettre image ici

# Les hyper-paramètres

- La capacité d'un modèle est déterminée par sa configuration (que l'on nomme hyper-paramètres)
- Ex. Un polynôme de degré  $k$  à la place d'une ligne.

Degré (hyper-p.)	Fonction de décision	Paramètres à apprendre
0	$f(x) = a$	$a$
1	$f(x) = ax + b$	$a, b$
2	$f(x) = ax^2 + bx + c$	$a, b, c$
etc..		

# Problème général : classifier une donnée selon ses caractéristiques

SUGGESTIONS D'EXEMPLE ALTERNATIF : Nombre d'heures de sommeil | Nombre d'heures de travail | Nombre d'heures passer à faire la fête | Occupation

	Alcool	Pourcentage	Type
1	Bière blonde	5%	Faible
2	Bière brune	9%	Faible
3	Vin	12%	Faible
4	Vodka	40%	Fort
5	Gin	47%	Fort
6	Rhum	55%	Fort
7	<i>Curaçao Bleu</i>	25%	?
8	<i>Whisky</i>	72%	?

---

# Qu'est-ce qu'on fait ici ?

- Nous allons vous montrer deux techniques de modèle non-paramétrés (non-paramétrés : les techniques n'apprennent pas à proprement parler de paramètres, elles ne font que garder en mémoire tous les exemples et calcule une réponse directement en fonction de ceux-ci)
- Nous allons ici tenter de classifier des couleurs en fonction de millions de données récoltés par sondage internet.



# Problème : Apprendre à nommer des couleurs

On cherche un algorithme qui peut nous donner le nom d'une couleur selon sa valeur `rgb`

Exemples

`rgb(255, 0, 0)` -> Rouge

`rgb(0, 255, 0)` -> Vert

`rgb(0, 0, 255)` -> Bleu

`rgb(0, 0, 0)` -> Noir

`rgb(255, 255, 255)` -> Blanc

`rgb(200, 80, 180)` -> ?

---

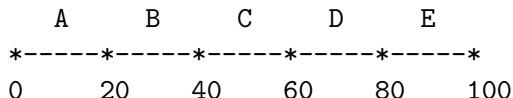
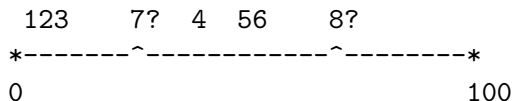
# Données : xkcd's color dataset

Le webcomic xkcd a récolté plus de XXXX échantillons de couleur étiquetée par des utilisateurs du web

Certains sont un peu abberants. Nous avons préalablement retirés toutes les couleurs de plus de YYYY caractères.

# Approche de l'histogramme : Séparer en catégories et trouver la tendance dans chaque catégorie

Séparer en 5 sur l'échelle de 0 à 100



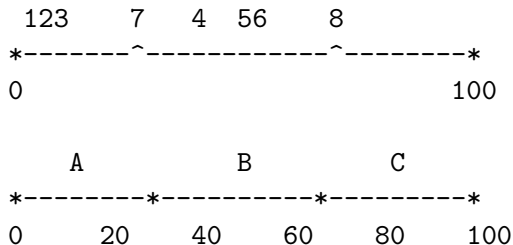
Catégories

A: Faible ; B: Fort ; C: Fort ; D: (vide) ; E: (vide)  
7 est donc Fort, 8 est ?

---

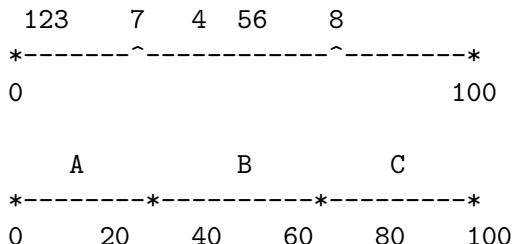
# Approche : Séparer en catégories et trouver la tendance dans chaque catégorie

Séparer en 3 sur l'échelle de 0 à 100



# Approche : Séparer en catégories et trouver la tendance dans chaque catégorie

Séparer en 3 sur l'échelle de 0 à 100



Catégories

A: Faible ; B: Fort ; C: Aucune donnée

7 est donc faible, 8 est ?

# Approche : Séparer en catégories et trouver la tendance dans chaque catégorie

## Avantages


- Très simple, semble suffisant dans certains cas

## Problèmes

- Certaines catégories peuvent être vides
    - Impossible de donner une réponse dans certains cas
  - Il faut trouver le nombre idéal de catégories
    - Pas assez de catégories ne donne pas une idée assez précise
    - Trop de catégories risque de donner beaucoup de cas où on ne sait pas répondre
-

## Autre approche : Les $k$ plus proches voisins

Trouver les  $k$  éléments les plus "proches" à ce qu'on cherche à identifier et déduire une catégorie en fonction de ces éléments (et potentiellement de



KnnClassification.png

leur distance) { width=30% }

## Autre approche : Les $k$ plus proches voisins

- Nécessite une définition de la distance entre 2 couleurs (distance euclidienne en 3 dimension dans notre cas)

Pour  $a$  et  $b$ , deux tableaux de nombre de taille 3, la distance se définit comme suit :

$$d(a, b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2}$$

- Nécessite une fonction de score. La catégorie choisie sera celle avec le plus haut score.

Les voisins :  $((x_1, y_1), (x_2, y_2), \dots, (x_k, y_k))$  (x la position, y la couleur)



Plusieurs variantes existent :

- Vote majoritaire des  $k$  plus proches voisins  
Score(Couleur) = Compte des couleurs de cette catégorie
- Vote pondéré des  $k$  plus proches voisins  
Score(Couleur, position) = Sommes de  $\frac{1}{\text{dist}(x_i, p)}$  pour tous les  $x_i$  tel que  $y_i$  est la couleur demandée

$$\text{score}(c, p) = \sum_{i=1}^k I_{c=y_i} \cdot \frac{1}{d(x_i, p)}$$

- Votre propre variante  
Inventez vos propres contraintes (e.g. une couleur doit être au moins présente 2 fois dans les  $k$  plus proches voisin pour pouvoir voter)

Vous vous rappelez des hyper-paramètres ? Ceux-ci contrôlent leur capacité à apprendre.

- Histogramme : le nombre de séparations dans chaque dimension
- KNN : le nombre de voisins

Mal ajustés, ils peuvent causer des réponses erronées.

Sur-apprentissage :    sous-apprentissage    bien ajusté