

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 ?

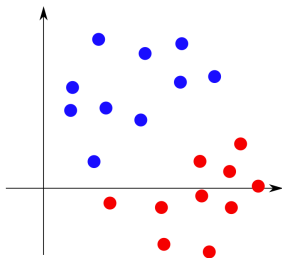
Champ d'étude de l'intelligence artificielle visant à apprendre à partir d'exemples les paramètres d'un modèle en vue d'accomplir une tâche.

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.



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

$x \in \mathbb{R}^4$ Fisher's Iris Data Y

Sepal Length	Sepal Width	Petal Length	Petal Width	Species ⁵⁷⁸
5.1	3.5	1.4	0.2	<i>setosa</i>
4.9	3.0	1.4	0.2	<i>setosa</i>
4.7	3.2	1.3	0.2	<i>setosa</i>
4.6	3.1	1.5	0.2	<i>setosa</i>
5.0	3.6	1.4	0.2	<i>setosa</i>
5.4	3.9	1.7	0.4	<i>setosa</i>
4.6	3.4	1.4	0.3	<i>setosa</i>
etc...				
5.7	2.8	4.5	1.3	<i>versicolor</i>
6.3	3.3	4.7	1.6	<i>versicolor</i>
4.9	2.4	3.3	1.0	<i>versicolor</i>
6.6	2.9	4.6	1.3	<i>versicolor</i>
5.2	2.7	3.9	1.4	<i>versicolor</i>
5.0	2.0	3.5	1.0	<i>versicolor</i>
etc...				
7.7	3.0	6.1	2.3	<i>virginica</i>
6.3	3.4	5.6	2.4	<i>virginica</i>
6.4	3.1	5.5	1.8	<i>virginica</i>
6.0	3.0	4.8	1.8	<i>virginica</i>
6.9	3.1	5.4	2.1	<i>virginica</i>
etc...				

$n=150$



Iris setosa

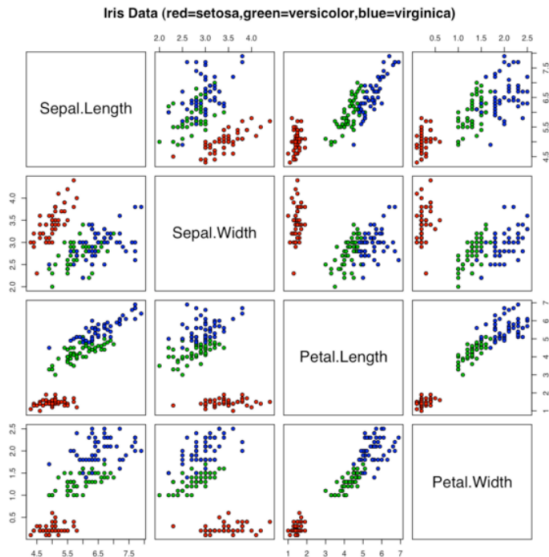


Iris versicolor



Iris virginica

Classifier une donnée selon ses caractéristiques



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 calculent 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

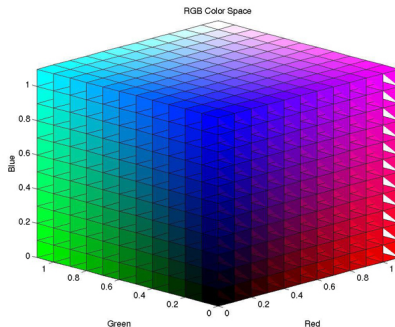
`rgb(200, 80, 180)` -> ?

- L'auteur du webcomic XKCD a récolté plus de 3 millions d'échantillons de couleur étiquetée par des utilisateurs du web
- Comme on peut s'y attendre, une certaine proportion des données est aberrante (lire *troll*).
- Chaque exemple est stocké sous forme de paires (rgb, étiquettes).

Approche : Séparer en sections et trouver la tendance

Technique de l'histogramme :

- Prendre l'espace d'entrée et le découper en sections de taille équivalente.



- Calculer la classe majoritaire des exemples dans chaque section et assigner sa valeur à la section.

Approche : Séparer en sections et trouver la tendance

Avantages

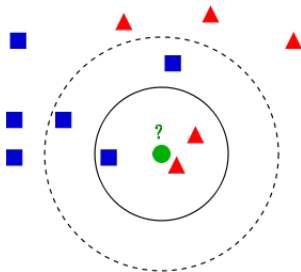
- Très simple, semble suffisant dans certains cas.

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
- La méthode est complètement inadaptée pour certaines tâches.
 - La prochaine méthode est beaucoup plus versatile et généralement plus efficace.

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 leur distance)



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

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.

Plusieurs variantes existent :

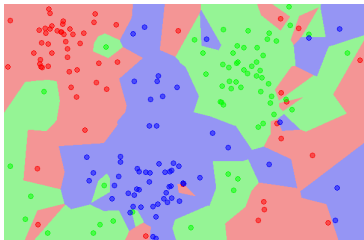
- Vote majoritaire des k plus proches voisins
Score(Couleur) = Compte des couleurs de cette catégorie
- Vote pondéré de tous les points dans l'ensemble
Score(Couleur, position) = Sommes de $\frac{1}{\text{dist}(x_i, p)}$ pour tous les x_i tel que y_i est la couleur demandée

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

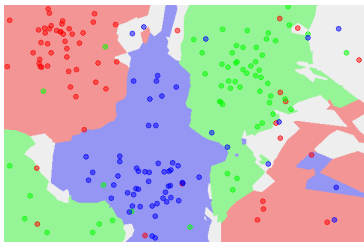
- Vote pondéré des k plus proches voisins
- Votre propre variante

Exemple de résultats

- 1 plus proche voisins, vote majoritaire :



- 5 plus proche voisins, vote majoritaire :



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

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

Mal ajustés, ils peuvent causer des réponses erronées et des aberrations. Trop apprendre peut être aussi dommageable que pas assez.

Des questions ?

Fin

Merci !