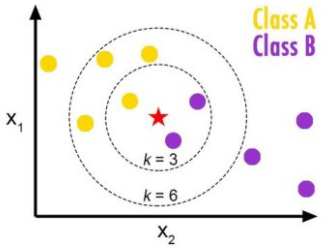
1NSI	Algorithmique k plus proches voisins	Cours – **Algorithme des k plus proches voisins**

Objectifs :

- ⇨ Introduire un algorithme basé sur l'apprentissage
- ⇨ Comprendre comment fonctionne l'algorithme des k plus proches voisins
- ⇨ Apprendre le vocabulaire associé à ce type d'algorithmes



I – Algorithme des k plus proches voisins (k NN)

L'algorithme des k plus proches voisins, souvent appelé **k NN** d'après son nom anglais **k N**earest **N**eighbors, est un algorithme d'**apprentissage automatique supervisé.**
Son fonctionnement peut être assimilé à l'analogie suivante « dis-moi qui sont tes voisins, je te dirais qui tu es ».

1) Classification

L'algorithme k NN peut être utilisé pour effectuer de la **classification** : un objet (au sens informatique, ce peut donc être n'importe quoi : un individu, un pays, un nombre …) sera associé à une catégorie en se basant sur les catégories auxquelles appartiennent ses k plus **proches** voisins (détermination du mode en statistiques)

Exemple 1 :
On peut tenter de prédire le département dans lequel se situe une commune à partir du département dans lequel se situent les k = 5 communes voisines les plus proches pour lesquelles cette information est connue (en considérant que c'est le cas au moins pour les préfectures et sous-préfectures).
Il existe un risque pour que la prédiction effectuée de cette manière ne soit pas correcte (en particulier pour une commune qui serait située le long de la frontière entre deux départements)

2) Régression, interpolation extrapolation

L'algorithme k NN peut aussi être utilisé pour effectuer de la **régression** : la valeur d'un paramètre numérique pour un objet peut être estimée à partir des valeurs de ce paramètre pour ses plus proches voisins.

Exemple 2 :
On peut tenter de prédire la taille et la masse à l'âge adulte d'une fille à partir des tailles et masses moyennes de sa mère et de ses deux grands-mères, qui sont ses k = 3 plus proches voisines dans l'arbre généalogique.

3) Apprentissage supervisé

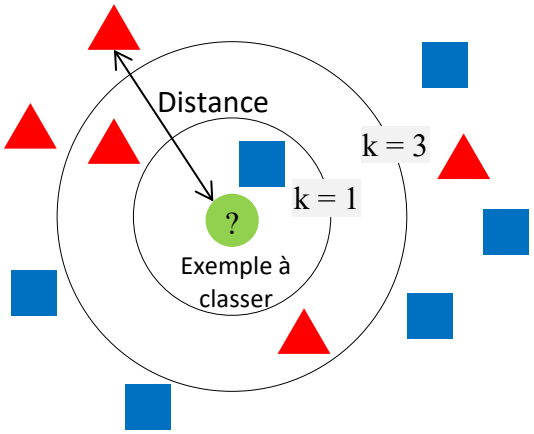
Pour que l'algorithme k NN fonctionne, il doit pouvoir se référer à un jeu de **données d'apprentissage** (contenant des informations sur les voisins potentiels) à partir desquelles il « apprend » automatiquement à répondre. Cet apprentissage est dit « supervisé », car les catégories ou valeurs de paramètres des objets faisant partie des données d'apprentissage sont fournies, ce qui permet un « entrainement sous contrôle » sur des exemples pour lesquels la réponse est connue.

Exemple 3 :
Pour tenter d'apprendre à un logiciel à reconnaître un arbre selon le principe de l'algorithme k NN, on pourrait imaginer lui fournir les 5 photographies ci-dessous.
La supervision prend la forme de la deuxième ligne dans laquelle chaque image reçoit une étiquette « arbre » ou « pas arbre ». Elle est indispensable pour un apprentissage par cette méthode.

				
Arbre	*Arbre*	*Pas arbre*	*Arbre*	*Pas arbre*

4) Distance

Une fois qu'il dispose des données d'apprentissage, l'algorithme k NN peut être appliqué à des **données de test** constituées d'objets pour lesquels il tente d'effectuer une prédiction (pas nécessairement correcte) en se référant aux données d'apprentissage pour les voisins les plus proches.
Cette notion de proximité implique d'être capable de mesurer une **distance** entre deux objets. Cette distance peut correspondre à la distance euclidienne (distance habituelle utilisée en géométrie) ou à une autre distance dans un sens mathématique plus abstrait.



Exemples :
Dans l'exemple 1, la proximité entre 2 villes sera mesurée par leur distance en kilomètres, on utilise donc la distance euclidienne.
Dans l'exemple 2, la proximité entre 2 individus peut être mesurée par le nombre de branches qui les séparent dans l'arbre généalogique.
Dans l'exemple 3, il est difficile de définir une distance adaptée permettant de dire que 2 images sont proches et on ne peut donc pas appliquer simplement un algorithme de type k NN. Cela explique l'utilisation des reconnaissances d'images dans les tests CAPTCHA[1] qui permettent de distinguer un robot et un utilisateur humain (utilisé pour contrôler l'accès à certains sites internet)

5) Prédicteurs et étiquettes

On distingue deux types d'informations sur les objets :

- les **prédicteurs**, qui permettent de les comparer en calculant une distance afin de dire s'ils sont proches ;
- les **étiquettes** qui correspondent aux catégories lorsque qu'on utilise un algorithme k NN de classification ou aux valeurs qu'on cherche à estimer lorsqu'on utilise un algorithme k NN de régression.

Exemples :
Dans l'exemple 1, on peut prendre comme prédicteurs les coordonnées GPS de la mairie, tandis que l'étiquette correspond au département.

[1] **C**ompletely **A**utomated **P**ublic **T**uring test to tell **C**omputers and **H**umans **A**part

Dans l'exemple 2, le prédicteur peut être la position dans l'arbre généalogique et les étiquettes sont la taille et la masse à l'âge adulte.
Dans l'exemple 3, l'étiquette est « arbre » ou « pas arbre ». Le prédicteur, mal défini, pourrait être la forme.

Les prédicteurs apparaissent à la fois dans les données d'apprentissage et dans les données de test (puisqu'il faudra calculer la distance entre les objets sur lesquels on teste l'algorithme et les objets de référence des données d'apprentissage.)
Les étiquettes sont disponibles pour les données d'apprentissage (c'est pour cela qu'on parle d'apprentissage supervisé), mais elles ne sont normalement pas disponibles pour les données de test (l'intérêt de l'algorithme k NN est justement de combler ce manque)

6) Choix de la valeur de k

Le choix de la valeur de k est un compromis qui dépend du type d'utilisation envisagé.
Plus k est grand, plus on se réfère à un grand nombre de données d'apprentissage. Intuitivement, la fiabilité et la stabilité du résultat renvoyé par l'algorithme k NN augmentent lorsque k augmente.
Cela est vrai, mais seulement jusqu'à un certain point. En effet, en augmentant trop la valeur de k, on en vient à considérer des « voisins » qui sont en réalité très éloignés de l'objet observé. La prise en compte des étiquettes de ces « voisins lointains » a alors tendance à fausser la prédiction.

Exemples :
Dans l'exemple 1, en prenant k = 35000, l'ensemble des k plus proches voisins de n'importe quelle commune sera l'ensemble de toutes les communes de France (puisqu'il y en a environ 35000), par conséquent, l'algorithme répondra pour n'importe quelle commune que son département est le « Pas-de-Calais » (département français comportant le plus de communes, soit environ 900), ce qui donnera donc une réponse fausse dans l'immense majorité des cas. Ce qui est logique puisqu'il n'est clairement pas pertinent de considérer que les communes du Pas-de-Calais font partie des plus proches voisines de la commune de Marseille !
Dans l'exemple 2, si on prend $k = 2^{15} - 1 = 65535$, il est clair que les informations sur des personnes ayant un lien de parenté au quinzième degré sont assez peu pertinente pour estimer la taille de la personne considérée. Inversement, prendre k = 1, revient à ne prendre en considération que la taille de la mère pour estimer la taille de la fille, ce qui ignore clairement les contributions génétiques de la branche paternelle.

En pratique, on effectue donc des essais avec différentes valeurs de k jusqu'à obtenir un compromis qui semble satisfaisant.

7) Complexité et terminaison

Si n désigne le nombre d'objets appartenant aux données d'apprentissage, alors la complexité de l'algorithme k-NN est O(n).
En effet, l'algorithme calcule la distance de l'objet dont un cherche l'étiquette avec chacun des n objets des données d'apprentissage (boucle bornée « for » de complexité O(n)) garde les k plus proches et calcule leur mode ou leur moyenne (boucle bornée « for » de complexité O(k))
Comme on cherche les voisins parmi les données d'apprentissage, on a $k < n$ donc la complexité globale de l'algorithme est $O(n)+O(k)=O(n)$. Même dans le cas k = n, on obtient $O(n)+O(n)=O(n)$

Il n'y a aucun doute sur la terminaison de l'algorithme, puisqu'on effectue uniquement des boucles bornées.

8) Apprentissage automatique, vraiment ?

Nous avons considéré l'algorithme k NN comme un algorithme d'apprentissage automatique.
Cependant, cet aspect ne fait pas l'unanimité.

En effet, cet algorithme ne « construit » pas de connaissance, il se contente de conserver toutes les données d'apprentissage contrairement à d'autres algorithmes d'apprentissage qui élaborent des règles de décisions en sélectionnant, pondérant et hiérarchisant des critères de décision.

Références :

Références conseillées :
http://isnbreizh.fr/1nsi/activity/algoRefKnn/index.html
https://mrmint.fr/introduction-k-nearest-neighbors
https://openclassrooms.com/fr/courses/4011851-initiez-vous-au-machine-learning/4022441-entrainez-votre-premier-k-nn

Référence pour aller plus loin (en anglais) :
https://scikit-learn.org/stable/