Examen 1ère session (2h) - 28 mai 2018

Rappels : **Tous documents autorisés.** Les calculatrices et autres appareils électroniques doivent être éteints et rangés. Le barème (sur 20) n'est donné qu'à titre indicatif.

Exercice 1 *Questions de cours (4 points)*
On considère une base d'apprentissage $\mathcal{X}$ contenant n exemples décrits par p attributs.
Les questions suivantes sont indépendantes.

Question 1. On considère d_E, la distance euclidienne. Soit deux exemples x_1 et x_2 de $\mathcal{X}$, donner l'expression de $d_E(x_1, x_2)$ en fonction des coordonnées de x_1 et de x_2.

Question 2. On considère maintenant le cas général d'une mesure de distance d quelconque. Soit $A = \{x_1, x_2, ..., x_{n_A}\}$ et $B = \{y_1, y_2, ..., y_{n_B}\}$ deux groupes d'exemples de $\mathcal{X}$, donner l'expression de la distance entre A et B pour chacune des approches de chaînage vues en cours ("simple linkage", "complete linkage", "average linkage" et "centroid linkage").

Question 3. On note $\mathcal{X} = \{x_1, \ldots, x_n\}$. Soit E et F deux sous-ensembles de $\mathcal{X}$ définis par leurs fonctions caractéristiques χ_E et χ_F.

1. donner l'expression de χ_E et de χ_F.
2. donner les fonctions caractéristiques de $E \cup F$ et de E^c (le complémentaire de E).
3. donner la définition d'une partition P de $\mathcal{X}$ en k sous-ensembles distincts.

Question 4. Qu'est-ce qui différencie l'algorithme des k-moyennes et l'algorithme des k-médoides ?

Question 5. Quels sont les avantages et les désavantages de l'algorithme des k-moyennes vis-à-vis de l'algorithme de classification hiérarchique ?

Exercice 2 *Apprentissage non-supervisé (6 points)*
On considère la base d'apprentissage de $[0, 10] \times [0, 10]$ contenant les 7 exemples suivants : $\mathcal{X} = \{(1, 2), (1, 4), (3, 4), (3, 5), (6, 2), (6, 5), (8, 3)\}$ (remarque : cette base est déjà normalisée).

Question 1. En détaillant les étapes et en expliquant les calculs réalisés et les regroupements effectués, appliquer sur $\mathcal{X}$ l'algorithme de classification hiérarchique, version ascendante, en utilisant l'approche "centroid linkage" et la distance euclidienne [1].

Question 2. Donner le dendrogramme obtenu par l'application de l'algorithme dans la question 1.

Question 3. On rajoute l'exemple $(5, 5)$ dans $\mathcal{X}$, sans ré-appliquer l'algorithme donner, en les justifiant, les modifications apportées au dendrogramme précédent par l'ajout de cet exemple.

Question 4. On décide d'appliquer l'algorithme des k-moyennes avec $k = 2$ sur la base $\mathcal{X}$ initiale (sans l'exemple $(5, 5)$). On choisit les exemples $e_1 = (1, 4)$ et $e_2 = (6, 2)$ pour initialiser les centres des clusters. Représenter graphiquement $\mathcal{X}$ et la frontière de séparation des 2 clusters induits par e_1 et e_2.

Question 5. Donner les coordonnées des 2 centres des deux clusters induits par e_1 et e_2.

Question 6. Donner l'inertie intra-cluster des clusters obtenus à la question précédente et en déduire l'inertie globale de la partition.

1. pour simplifier les calculs, ne pas calculer la racine carrée.

Question 7. Au bout de combien d'étapes l'algorithme des k-moyennes converge-t-il pour cette initialisation par e_1 et e_2 ?

Question 8. L'algorithme des k-moyennes est sensible au choix des centres initiaux, quel choix d'exemples initiaux dans $\mathcal{X}$ entraînerait une convergence en un grand nombre d'étapes lors de l'application sur $\mathcal{X}$ avec $k = 2$?

Exercice 3 *Perceptron (3 points)*
On considère un perceptron simple avec deux entrées notées x_1 et x_2 et une sortie y telle que :

$$y = \begin{cases} 1 \text{ si } w_1x_1 + w_2x_2 - w_0 > 0 \\ 0 \text{ sinon} \end{cases}$$

Question 1. Trouver les poids pour que le perceptron calcule la fonction ET logique.

Question 2. Même question avec la fonction OU logique.

Question 3. Essayer de trouver des poids pour la fonction XOR et commenter.

Question 4. Construire un réseau de neurones qui calcule la fonction XOR.

Exercice 4 *Kernel trick (4 points)*
Soit un problème à 2 classes ω_1 et ω_2. On dispose des points d'apprentissage suivants pour les classes :

$$\omega_1 = \{(2,1),(1,3),(2,3)\} \text{ et } \omega_2 = \{(1,0),(0,1),(-1,1)\}.$$

On considère un classifieur à correction d'erreur qui parcourt les points dans l'ordre de façon séquentielle et qui modifie le vecteur normal dès que l'on rencontre un point mal classé. Après une correction, on recommence avec le premier point d'apprentissage. Le vecteur initial vaut $(w_1 = 1, w_2 = 2, w_3 = 0)$, le modèle calculant en sortie le signe de $w_1\,x_1 + w_2\,x_2 + w_3$.

Question 1. Expliquer en quoi consiste le *kernel trick* et comment celui-ci s'applique dans un espace en 2 dimensions afin d'autoriser l'hyperplan séparateur à ne pas passer obligatoirement par l'origine $(0,0)$.

Question 2. Dérouler l'algorithme du perceptron. En combien d'itérations l'algorithme converge-t-il ? (remarque : inférieur à 15)

Question 3. Calculer l'équation des frontières de décision à l'issue de l'application de l'algorithme et en déduire la règle de décision pour chacune des classes.

Question 4. Que se passe-t-il si on mélange les exemples en alternant les exemples de chaque classe ?

Question 5. Tracer sur une même figure les points d'apprentissage et la surface de décision.

Exercice 5 *(3 points)*
Vous avez trois mois devant vous. Décrivez une application d'apprentissage que vous aimeriez réaliser (champ libre, allant de l'apprentissage pour les jeux vidéo au robot chauffeur) : quel est le but ? pourquoi utiliser de l'apprentissage ? comment récolter les données ? comment les pré-traiter ? quel algorithme choisir ? comment valider les résultats ?