ISM MP, Informatique
Année 2025/2026

# TP Informatique 15

## Exercice 1

Soit $k$ un entier non nul et $\mathscr{A} = (\mathrm{X}_i, c_i)_{1\leqslant i\leqslant n}$ une base d'apprentissage avec $\mathrm{X}_i \in \mathbb{R}^d$ et $c_i \in [\![ 1\,;\,\mathrm{C} ]\!]$ la classe du point $\mathrm{X}_i$ pour tout $i \in [\![ 1\,;\,n ]\!]$. L'algorithme des $k$ *plus proches voisins* associe à un nouveau point $\mathrm{X} \in \mathbb{R}^d$ la classe majoritaire des $k$-éléments de la famille $(\mathrm{X}_i)_{1\leqslant i\leqslant n}$ les plus proches de X au sens de la métrique euclidienne dans $\mathbb{R}^d$.

Il consiste en les étapes suivantes :
- Saisie d'un nouveau point $\mathrm{X} \in \mathbb{R}^d$ ;
- Calculer $d_i = \|\mathrm{X} - \mathrm{X}_i\|$ pour tout $i \in [\![ 1\,;\,n ]\!]$ ;
- Trier par ordre croissant la liste $[d_1, \ldots, d_n]$ en $\left[d_{\sigma(1)}, d_{\sigma(2)}, \ldots, d_{\sigma(n)}\right]$ avec $\sigma$ permutation de $[\![ 1\,;\,n ]\!]$ telle que $d_{\sigma(1)} \leqslant d_{\sigma(2)} \leqslant \ldots \leqslant d_{\sigma(n)}$ ;
- Choix de la classe majoritaire dans la liste $\left[c_{\sigma(1)}, \ldots, c_{\sigma(k)}\right]$.

Exécuter le fichier `TP15_EX01.py`. Celui-ci affiche le nuage de points de la population d'apprentissage stockée dans la variable `pop_0` puis celui de la population de test stockée dans la variable `pop_1`.

1. Compléter la fonction `majo(L)` d'argument `L` une liste non vide qui renvoie un élément majoritaire dans celle-ci.
2. Compléter la fonction `knn(k,X,A)` d'arguments `k` un entier non nul, `X` un point de $\mathbb{R}^d$ et `A` une base d'apprentissage qui renvoie le résultat de l'algorithme des $k$ plus proches voisins. La liste `A` est une liste de C sous-listes où chaque sous-liste `A[c]` contient les points $\mathrm{X}_i$ au format `ndarray` de classe $c_i = c$. La fonction `alg.norm` calcule la norme euclidienne dans $\mathbb{R}^d$. La fonction `sorted(L,key=lambda u:u[1])` d'argument `L` une liste de couples renvoie la liste triée par ordre croissant selon le deuxième argument du couple.
3. Compléter la construction de la matrice de confusion pour la base de test `pop_1`.
4. Tester le programme. On pourra modifier à son gré le choix de `k`, le choix des moyennes et écarts-types servant à générer les populations d'apprentissage et de test, etc. ...

## Exercice 2

On se propose d'utiliser l'algorithme des $k$ plus proches voisins pour effectuer de la reconnaissance de caractères.

Enregistrer les fichiers `train-images.idx3-ubyte` et `train-labels.idx1-ubyte` puis exécuter le fichier `TP15_EX02.py`.

La variable `imagearray` est un tableau contentant 60 000 images de taille $28 \times 28$ et la variable `labelarray` est un tableau contenant les chiffres représentés sur les images de `imagearray`. Ainsi, pour `i` un entier dans $[\![ 0\,;\,59\,999 ]\!]$, l'appel `imagearray[i]` renvoie un tableau codant une image de taille $28 \times 28$ et `labelarray[i]` renvoie le chiffre représenté sur l'image.

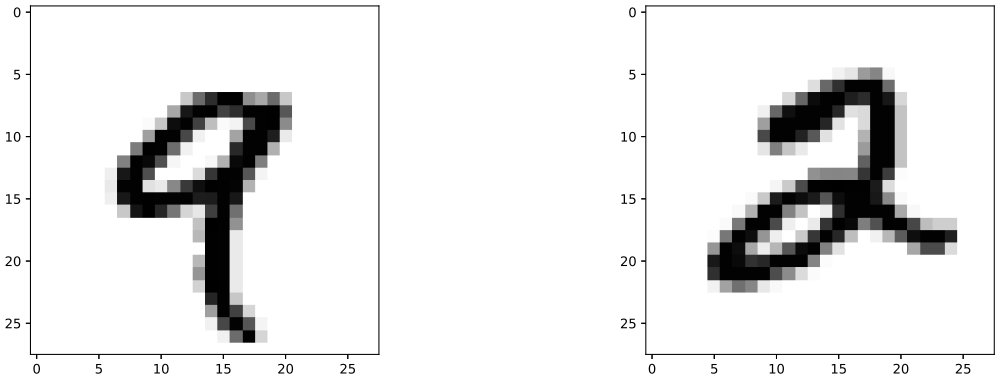

FIGURE 1 – Images d'indices 4 et 5 extraites de la base MNIST

1. Compléter les fonctions `majo(L)` et `knn(k,X,A)` comme dans l'exercice précédent.
2. Compléter la reconnaissance de caractères en faisant croître la taille de la base d'apprentissage et en calculant le taux de reconnaissance de caractères sur les 100 dernières images de la base `MNIST`. Commenter le graphe obtenu.