

## TME 2 - Classifieur bayésien, $k$ -plus proches voisins

### Données

En plus des données `imdb` du précédent TME, vous devrez tester vos algorithmes sur les données USPS du TME3 de MAPSI [http://webia.lip6.fr/~mapsi/uploads/Cours/2015\\_tme3\\_usps\\_train.txt](http://webia.lip6.fr/~mapsi/uploads/Cours/2015_tme3_usps_train.txt). Utilisez le code suivant pour lire le fichier de données :

```
1 def load_usps(filename):
2     with open(filename, "r") as f:
3         f.readline()
4         data = [ [float(x) for x in l.split()] for l in f if len(l.split())>2]
5         tmp = np.array(data)
6         return tmp[:,1:], tmp[:,0].astype(int)
7 datax, datay = load_usps("usps.txt")
```

Afin de visualiser les résultats, on utilisera également des jeux de données artificiels en 2d. Chargez l'archive `arftools.py`, elle contient trois fonctions à utiliser dans ce TME :

```
1 from arftools import *
2 # generation artificielle de donnees, data_type indique le type de jeux de donnees
   (0,1 ou 2)
3 datax, datay = gen_arti(data_type = 0, nbex = 1000) #1000 exemples de melange de deux
   gaussiennes
4 plot_data(datax, datay) # afficher les donnees
5 plot_frontiere(datax, f, step=20) # dessine la frontiere de decision en fonction de la
   fonction de prediction f
```

### Expérimentations

Coder :

- un classifieur bayésien.
- un algorithme des plus proches voisins.

Vous pouvez prendre modèle sur l'objet `Classifier` fourni dans l'archive. Quels paramètres pour le classifieur bayésien ? pour les plus proches voisins ?

Expérimentez sur les 3 jeux de données en validation croisée. Visualisez pour les jeux artificiels en 2d les frontières de décisions en fonction du nombre de voisins pour les  $k$ -nn.