TP4

IAA

Classification par K Plus Proches Voisins

1.Implémentation

Tout d'abord on ne dispose pas de 20 échantillons pour chaque classe comme indiqué dans le sujet mais bien de 19 échantillons, il vaut donc mieux prendre K <= (19*3 = 57)

```
In [7]: len(donnees_classe1)
Out[7]: 19

In [8]: len(donnees_classe2)
Out[8]: 19

In [9]: len(donnees_classe3)
Out[9]: 19
```

En prenant K=20 pour le point « donnee_test_classe1 »

(= [3.0299364237682695, 0.069933489299203899]), on a une indécision. En effet on a nbOccClasse = [10, 10, 0] , ce qui signifie 10 points dans la classe 1 et 10 points dans la classe 2.

Le cas où on ne prend pas de décision en cas d'égalité est la fonction prendPasDecision et a été mise en commentaire pour les exercices suivant. Dans ce cas j'ai levé une exception.

```
In [26]: run ./kppv.py
Decision par K-PPV, avec K =  20
nbOccClasse = [10, 10, 0]
nbOccClass = [10, 10, 0]
les classes qui portent soucis sont:[1 2]
---------------------------------------------------------------------------
Exception                                 Traceback (most recent call last)
...
Exception: indecision
```

2.Traitement du cas d'égalité

On peut prendre une fonction random qui va choisir au hasard une classe parmi celles qui

posent soucis.

On peut prendre une fonction qui va incrémenter ou décrémenter la valeur de K jusqu' à ce qu'il n'y ait plus d'indécision. Si la valeur de K est trop grande ou trop petite on aura une mauvaise décision.

On peut prendre une fonction qui va attribuer un poids pour chacun des points, ce poids peut être par exemple la distance (attention, plus une distance est petite plus le poids doit être important, on peut prendre 1/(distance+1) comme poids par exemple (pour éviter de diviser par 0). Attention dans ce cas il y a quand même une probabilité (très faible certes) pour qu'il y ait encore une indécision.

Les fonctions codées pour traiter ces cas sont prendDecisionRandom, DecisionIncrementeK, DecisionDecrementeK et DecisionPoidsDistance.

```
In [5]: run ./kppv.py
Decision par K-PPV, avec K =  20
nbOccClasse = [10, 10, 0]
La donnee de classe 1 a ete reconnue comme une donnee de classe  2
point a classer [3.0299364237682695, 0.0699334892992039]
```

Illustration 1: exemple d'un cas d'indécision résolu avec random

```
In [6]: run ./kppv.py
Decision par K-PPV, avec K =  20
nbOccClasse = [10, 10, 0]
indecision, on decremente K. K= 21
nbOccClasse = [10, 11, 0]
La donnee de classe 1 a ete reconnue comme une donnee de classe  2
point a classer [3.0299364237682695, 0.0699334892992039]
```

Illustration 2: exemple d'un cas d'indécision résolu en incrémentant K

```
In [7]: run ./kppv.py
Decision par K-PPV, avec K =  20
nbOccClasse = [10, 10, 0]
indecision, on decremente K. K= 19
nbOccClasse = [10, 9, 0]
La donnee de classe 1 a ete reconnue comme une donnee de classe  1
point a classer [3.0299364237682695, 0.0699334892992039]
```

Illustration 3: exemple d'un cas d'indécision résolu en décrémentant K

On remarque qu'on a eu une décision différente en incrémentant ou en décrémentant K. Ce point fait donc polémique.

```
In [8]: run ./kppv.py
Decision par K-PPV, avec K =  20
nbOccClasse = [10, 10, 0]
valeurs Classes [ 7.85999078  6.99022266  0.        ]
La donnee de classe 1 a ete reconnue comme une donnee de classe  1
point a classer [3.0299364237682695, 0.0699334892992039]
```

Illustration 4: exemple d'un cas d'indécision résolu en ajoutant un poids aux points choisi

3.Affichage

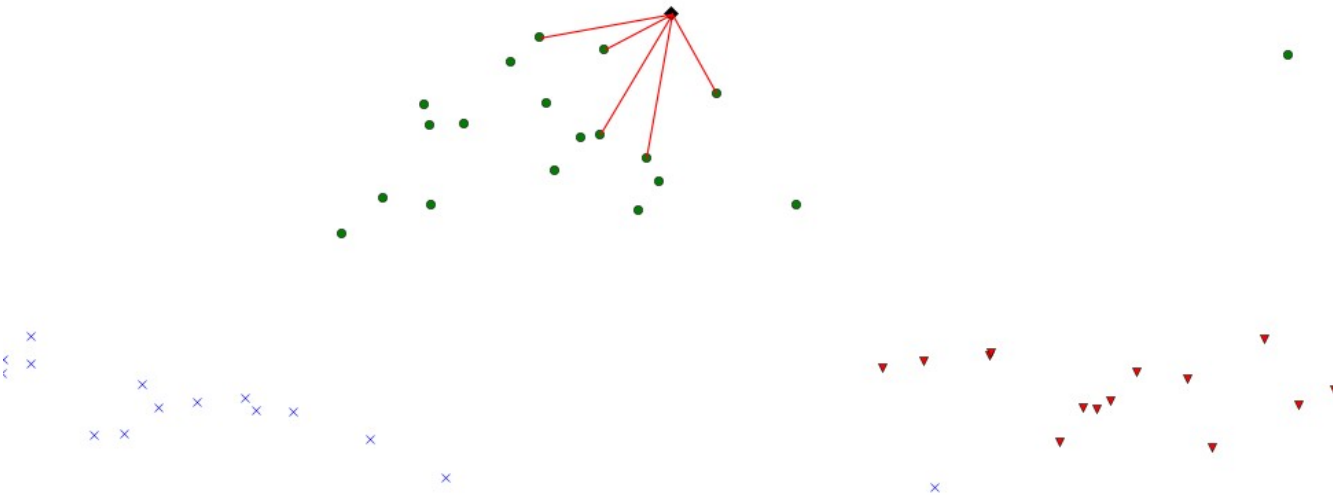

Illustration 5: affichage pour K=5 du point donnee_test_classe2 = [3.3156657804150163, 0.67327144428452557]

Attention pour pouvoir bien observer visuellement les vraies distances il faut penser à normaliser les axes du repère.

4.Choix des paramètres

Il ne faut pas prendre un K trop élevé car sinon on va prendre trop de points trop éloignés et la décision sera faussé.

Il est préférable de prendre un nombre de points représentatifs élevés, mais à condition qu'on prenne à peu près le même nombre de points représentatifs pour chaque classe (peut être un peu plus de points pour les classes plus « étalées »).

Ce qui va nous guider c'est la dimension et la taille des données, et leurs répartitions dans les différentes classes. (répartitions du point de vue taille des échantillons dans chaque classe et répartition dans « l'espace »).

5.Généralisation

On réutilise la fonction genere_donnee que j'avais codé au TP4 permettant de generer des données en dimension 3 sur 4 classes. Comme ces données sont générées aléatoirement elles ne donnent pas les même résultat.

Les distances se calculant grâce à la fonction norm, pas de problème de dimension.

```
5.Generalisation

point a classer :  [0.0, 0.0, 0.0]
nbOccClasse = [5, 0, 0, 0]
La donnee de classe 1 a ete reconnue comme une donnee de classe  1


point a classer :  [3.0, 3.0, 3.0]
nbOccClasse = [1, 4, 0, 0]
La donnee de classe 2 a ete reconnue comme une donnee de classe  2


point a classer :  [0.0, 0.0, 3.0]
nbOccClasse = [0, 0, 5, 0]
La donnee de classe 3 a ete reconnue comme une donnee de classe  3


point a classer :  [0.0, 3.0, 0.0]
nbOccClasse = [1, 1, 1, 2]
La donnee de classe 4 a ete reconnue comme une donnee de classe  4
```

Illustration 6: Pour K=5, cas sans indécision avec bonne prise de décision

```
5.Generalisation

point a classer :  [0.0, 0.0, 0.0]
nbOccClasse = [2, 0, 1, 2]
valeurs Classes [ 1.11646534  0.          0.53735811  1.20379476]
La donnee de classe 1 a ete reconnue comme une donnee de classe  4


point a classer :  [3.0, 3.0, 3.0]
nbOccClasse = [1, 3, 1, 0]
La donnee de classe 2 a ete reconnue comme une donnee de classe  2


point a classer :  [0.0, 0.0, 3.0]
nbOccClasse = [2, 1, 2, 0]
valeurs Classes [ 1.23075285  0.5574539   1.03305207  0.        ]
La donnee de classe 3 a ete reconnue comme une donnee de classe  1


point a classer :  [0.0, 3.0, 0.0]
nbOccClasse = [1, 0, 1, 3]
La donnee de classe 4 a ete reconnue comme une donnee de classe  4
```

Illustration 7: pour k=5 cas où il y a 2 cas d'indécision et deux mauvaises décisions pour chacun

```
5.Generalisation

point a classer :  [0.0, 0.0, 0.0]
nbOccClasse = [1, 0, 0, 4]
La donnee de classe 1 a ete reconnue comme une donnee de classe  4


point a classer :  [3.0, 3.0, 3.0]
nbOccClasse = [0, 4, 1, 0]
La donnee de classe 2 a ete reconnue comme une donnee de classe  2


point a classer :  [0.0, 0.0, 3.0]
nbOccClasse = [0, 3, 2, 0]
La donnee de classe 3 a ete reconnue comme une donnee de classe  2


point a classer :  [0.0, 3.0, 0.0]
nbOccClasse = [0, 0, 1, 4]
La donnee de classe 4 a ete reconnue comme une donnee de classe  4
```

Illustration 9: pour K=5 mauvaise décision sans indécision

Contact

Si il y a des erreurs, des remarques, des ajouts à faire, etc.

Veuillez m'en faire part à cette adresse :

wedg@hotmail.fr