
Rapport

Introduction à l'apprentissage

Table des matières

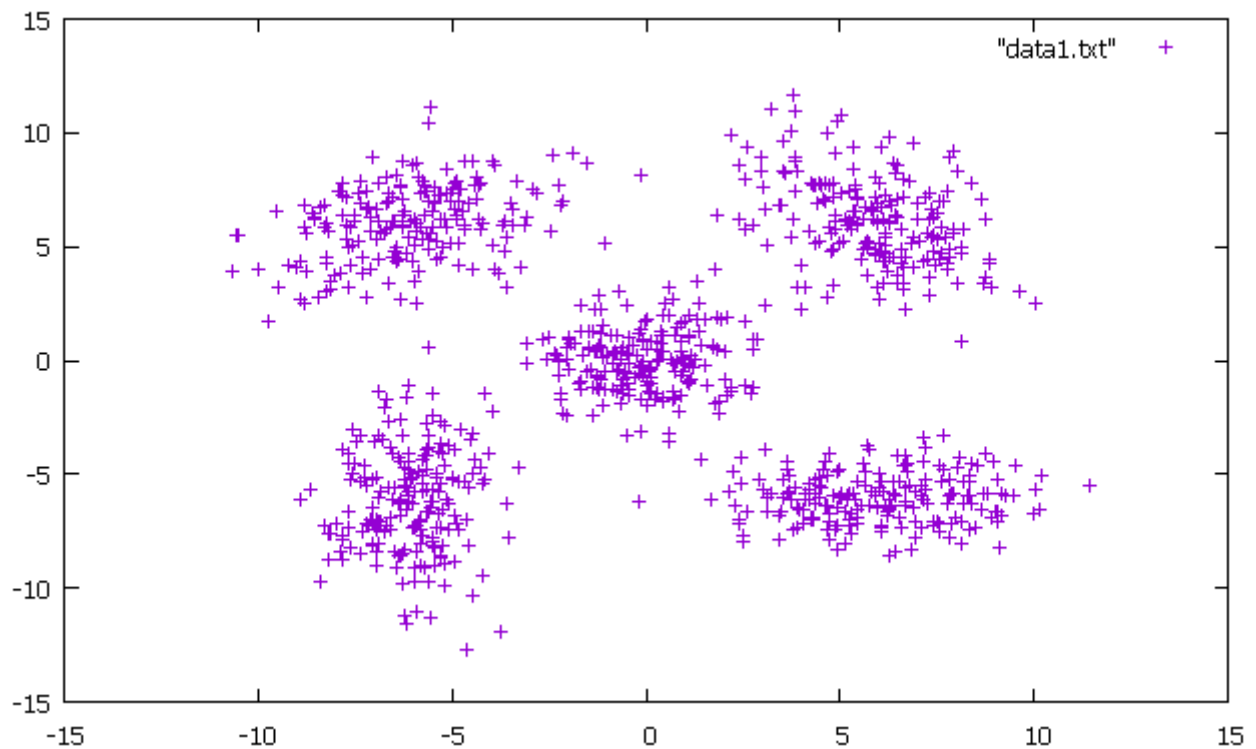
Classification des données bidimensionnelles.....	3
Observations.....	3
Data1 :.....	3
Data2 :.....	4
Data3 :.....	5
Première approche : les k-means.....	6
Data 1 :.....	7
Data 2 :.....	8
Data 3 :.....	9
Carte de Kohonen :.....	10
Data 1 :.....	10
Data 2 :.....	11
Data 3 :.....	12
Étude des données du sonar.....	13
Classification.....	13
Discrimination.....	13
Spike-Timing-Dependent Plasticity.....	15
Classification de signaux de parole.....	15

Classification des données bidimensionnelles

Observations

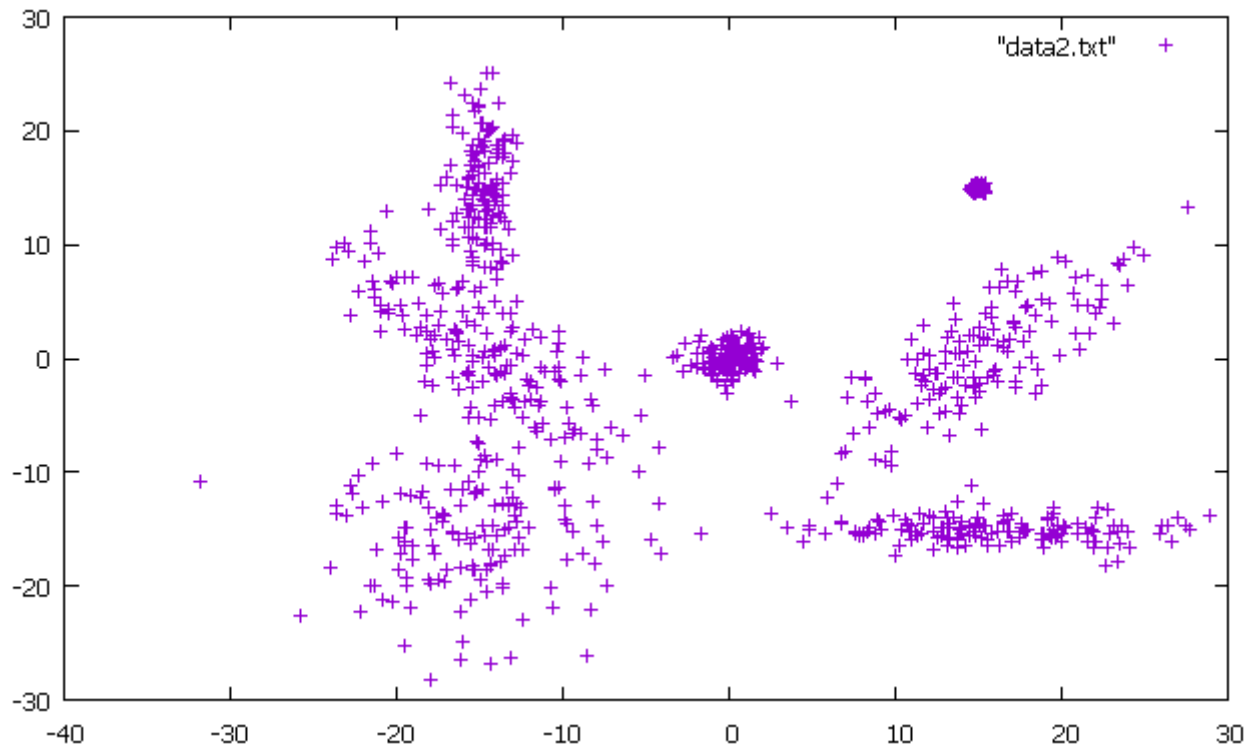
Les données des quatre fichiers appartiennent à R. Celles de data1 et data3 semblent être comprises entre -10 et 10 alors que celles de data2 entre -30 et 30. Grâce au logiciel de visualisation gnuplot nous pouvons facilement afficher les nuage de points que forment les données dans un repère cartésien.

Data1 :



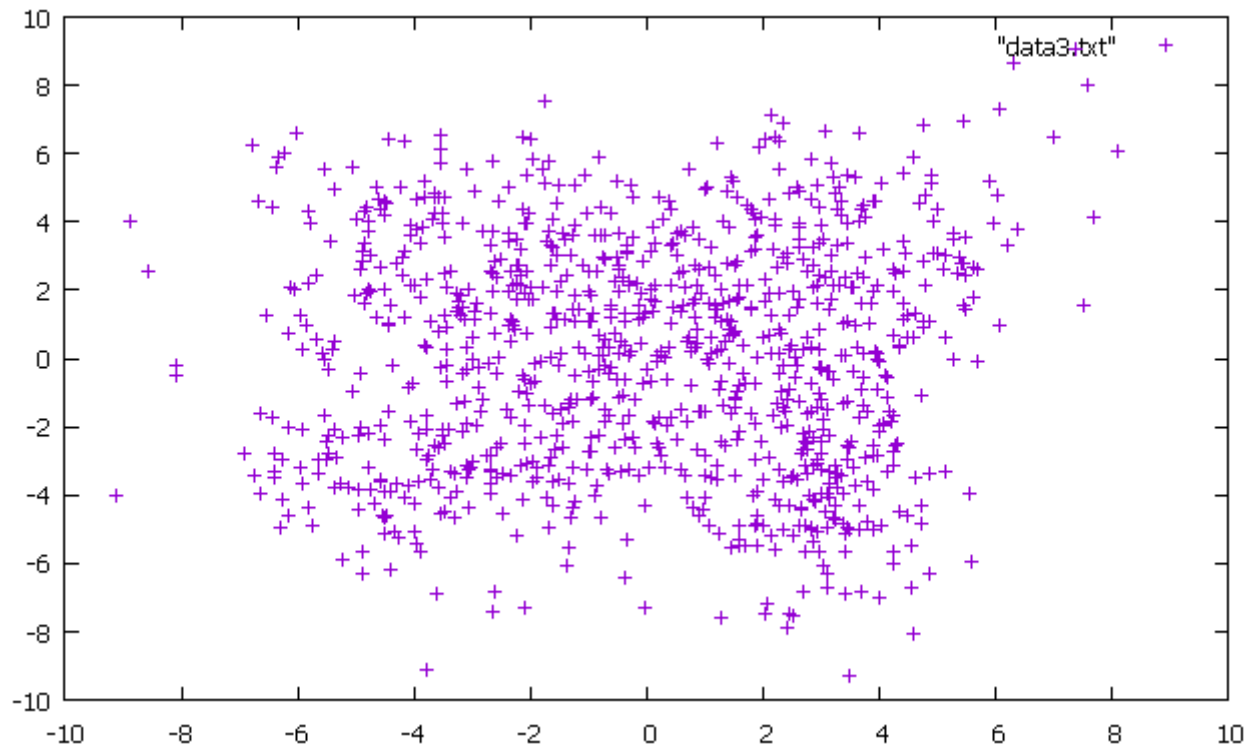
On distingue aisément cinq nuages de points qui ne se recouvrent pas. La classification ne devrait donc pas être trop difficile.

Data2 :



Ici on peut observer des classes de formes différentes avec peut être du recouvrement sur la partie gauche, nous préconisons donc l'utilisation des nuées dynamiques, ou des centres mobiles flous.

Data3 :



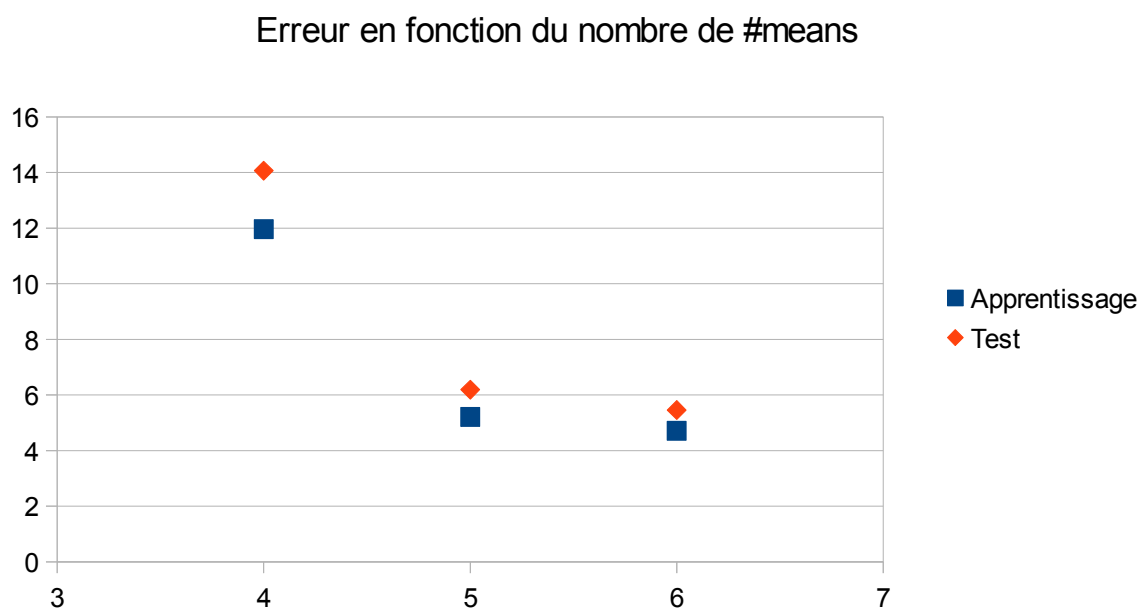
Sur le graphe du troisième jeu de donnée, nous ne voyons qu'un gros nuage de points homogène où il semble bien difficile de faire de la classification.

Les données de data4d sont décrites par quatre réels et ne sont donc pas objectivement observable dans un espace de taille 2 voir 3.

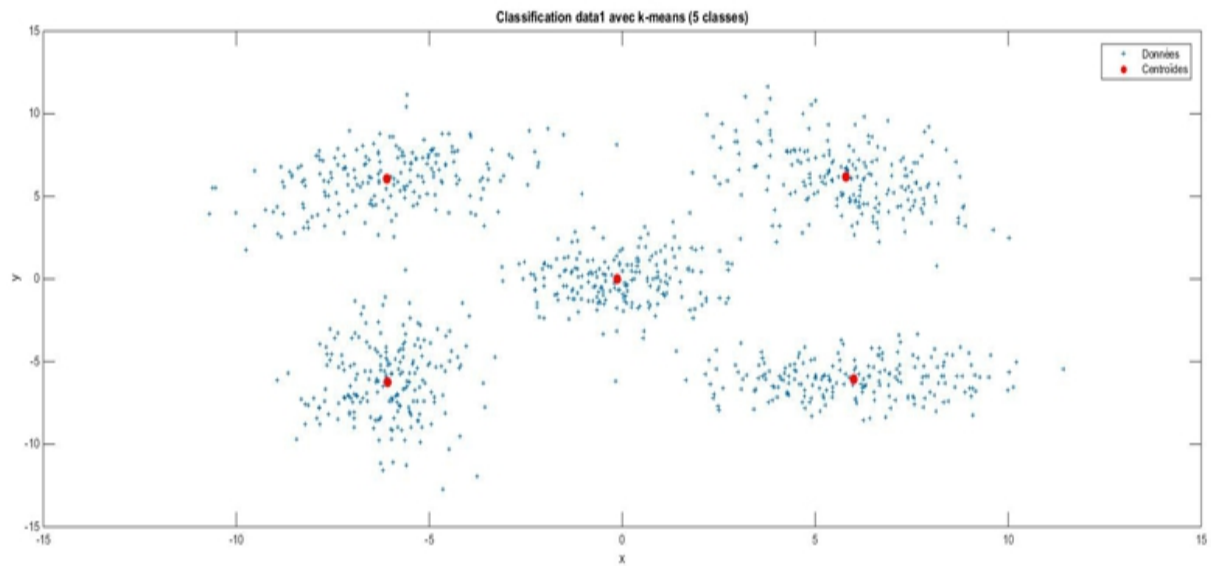
Première approche : les k-means

Pour les trois fichiers de données, nous utiliserons 66 % du corpus pour l'apprentissage et le reste pour les tests.

Data 1 :



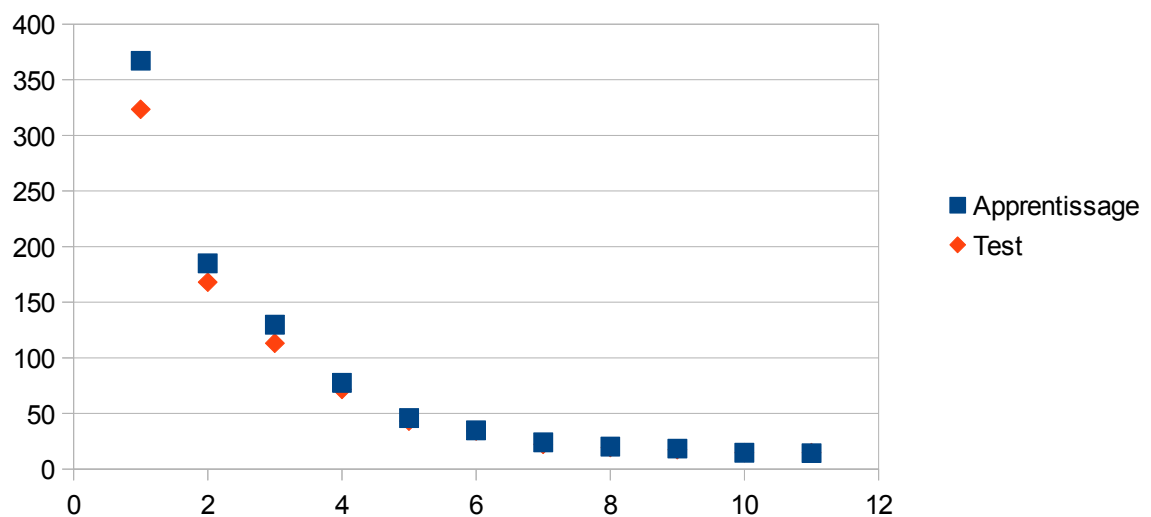
On voit bien que le nombre de classe à utiliser pour les k-means est bien de 5. En effet, l'erreur d'apprentissage diminue largement en passant de 4 à 5 classes mais très peu en passant de 5 à 6 classes.



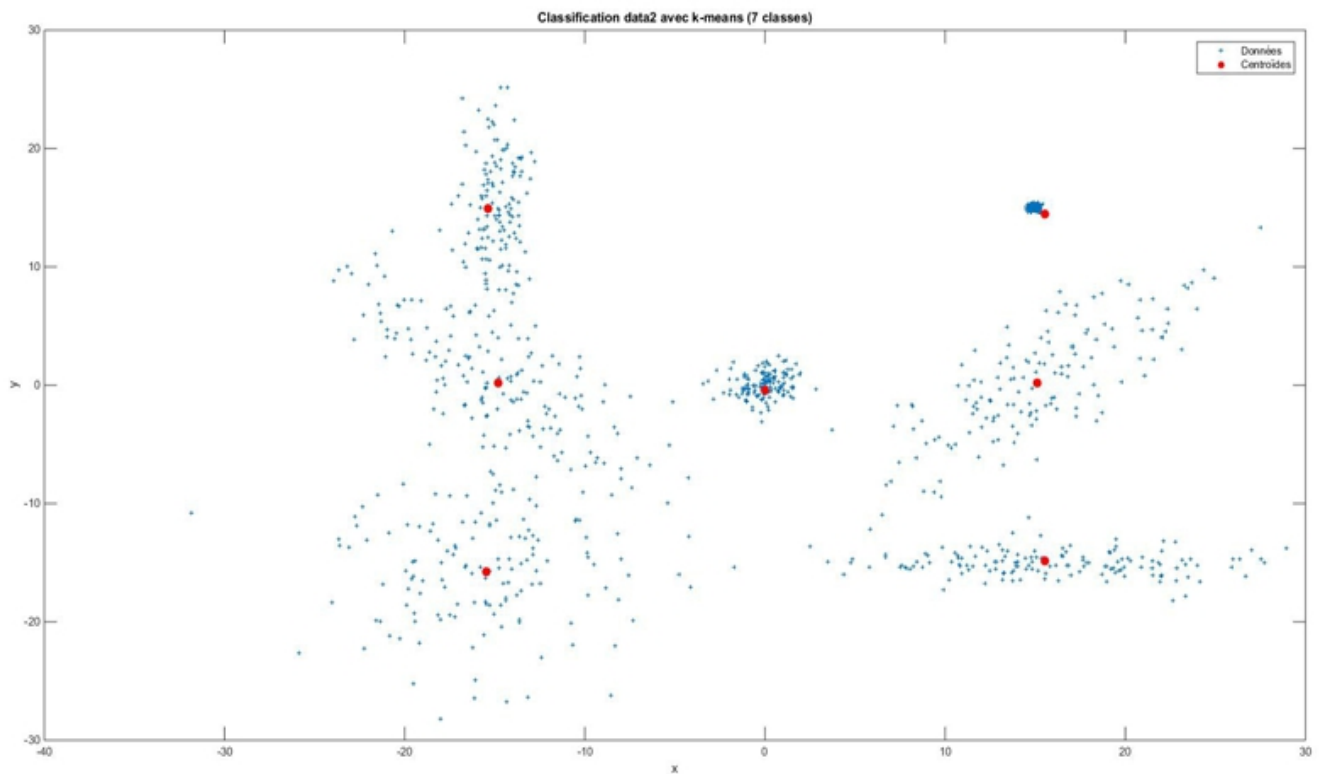
On voit bien ici les 5 centres de chaque classe sur les 5 différents nuages de points.

Data 2 :

Erreur en fonction du nombre de #means



Sur cette courbe, on voit que le nombre de classe à utiliser pour les k-means est de 7.

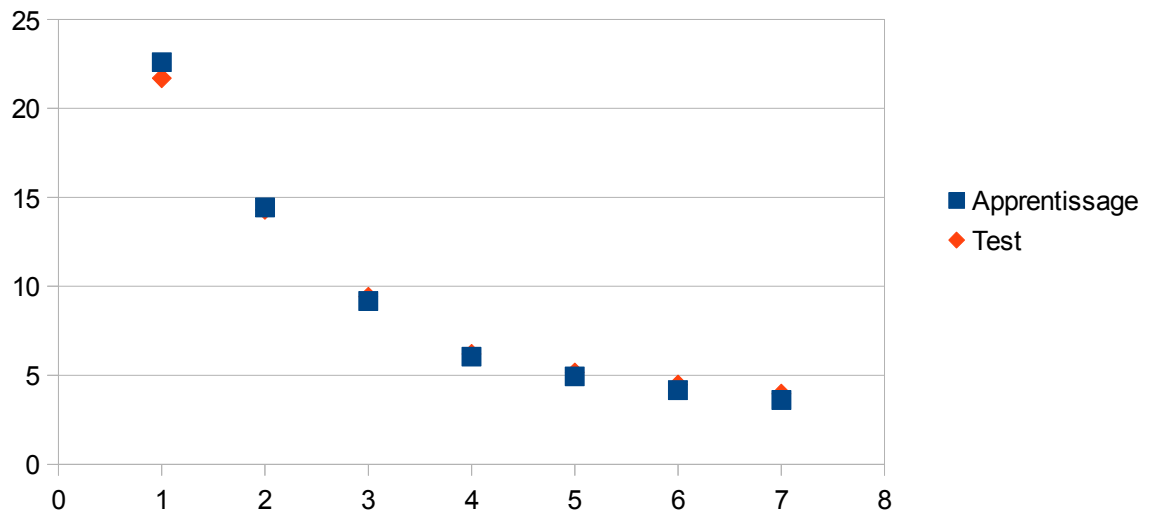


Sur cette courbe, on voit que le nombre de classes à utiliser pour les k-means est de 7.

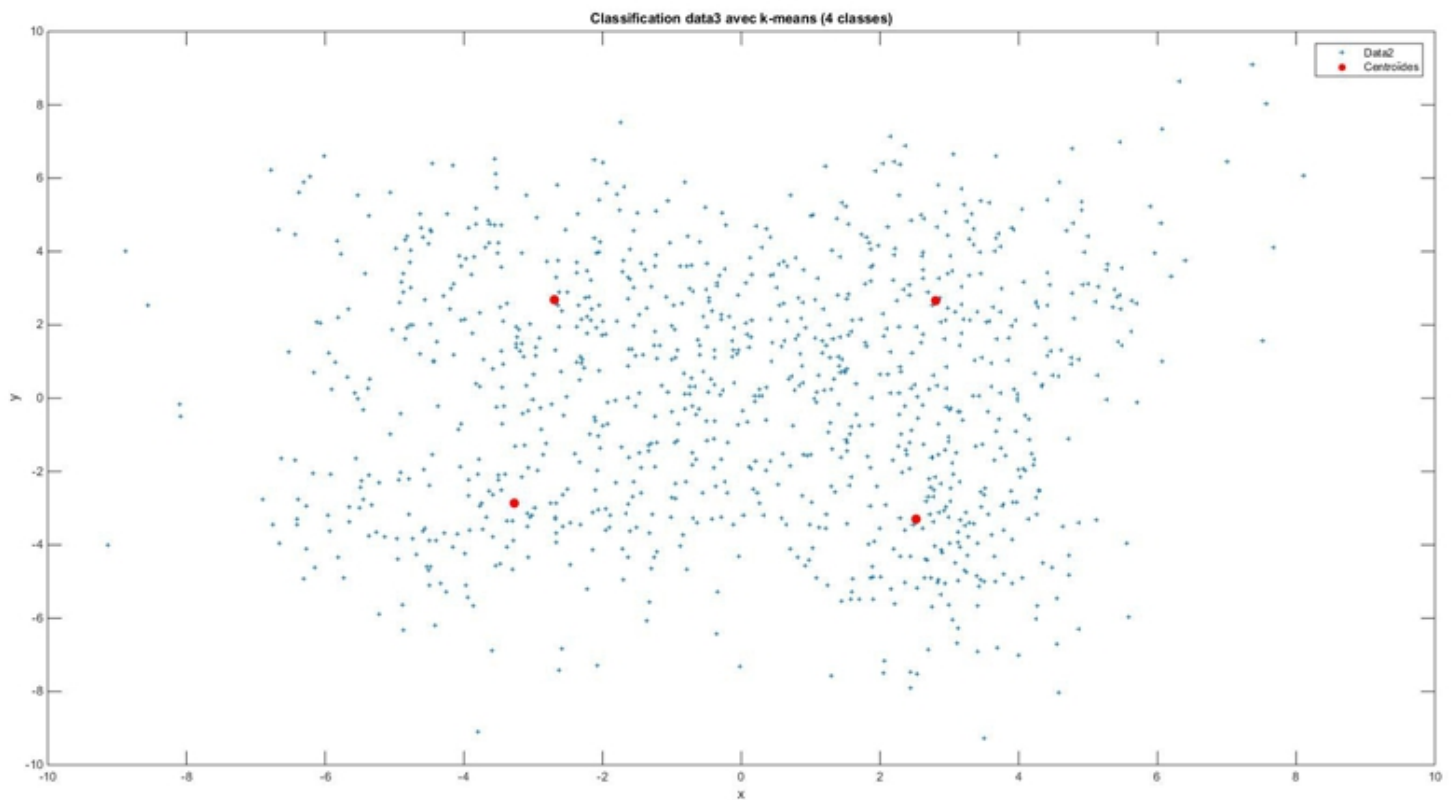
Ici 4 centres se trouvent naturellement sur les 3 nuages de droite et celui du milieu. Puis les 3 derniers centres se répartissent sur l'amas de gauche où il y a probablement recouvrement.

Data 3 :

Erreur en fonction du nombre de #means



On voit que pour data3, il faut utiliser 4 classes.

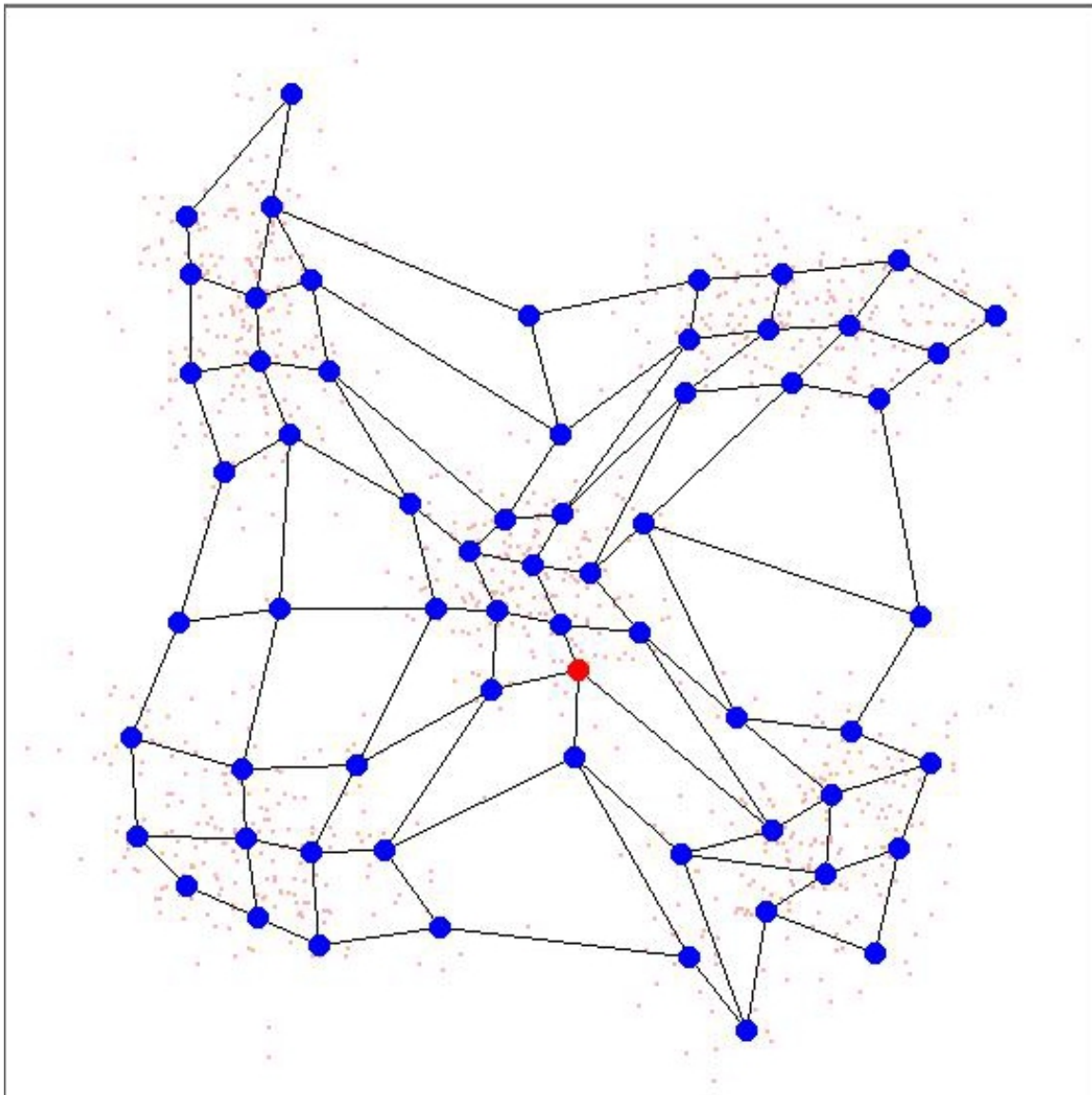


Les 4 centres se répartissent sur l'ensemble des points.

Carte de Kohonen :

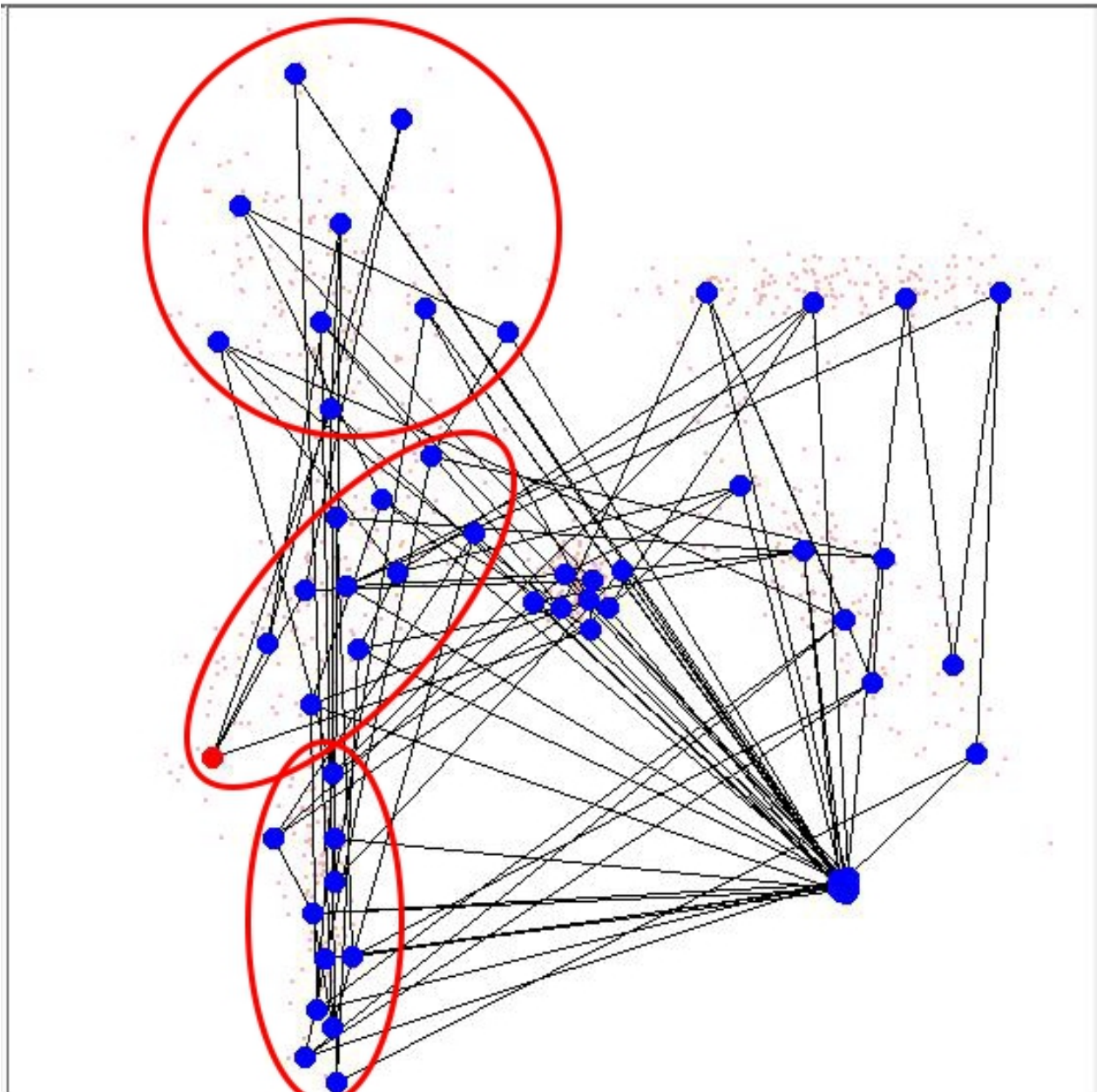
Grâce aux résultats de la classification par carte auto-organisatrice de Kohonen, nous avons une vue d'ensemble de la classification à obtenir. On peut donc savoir à l'avance le nombre de classes à utiliser et où placer ses points d'initialisation. Car rappelons-le, les k-means sont très sensibles à l'initialisation.

Data 1 :



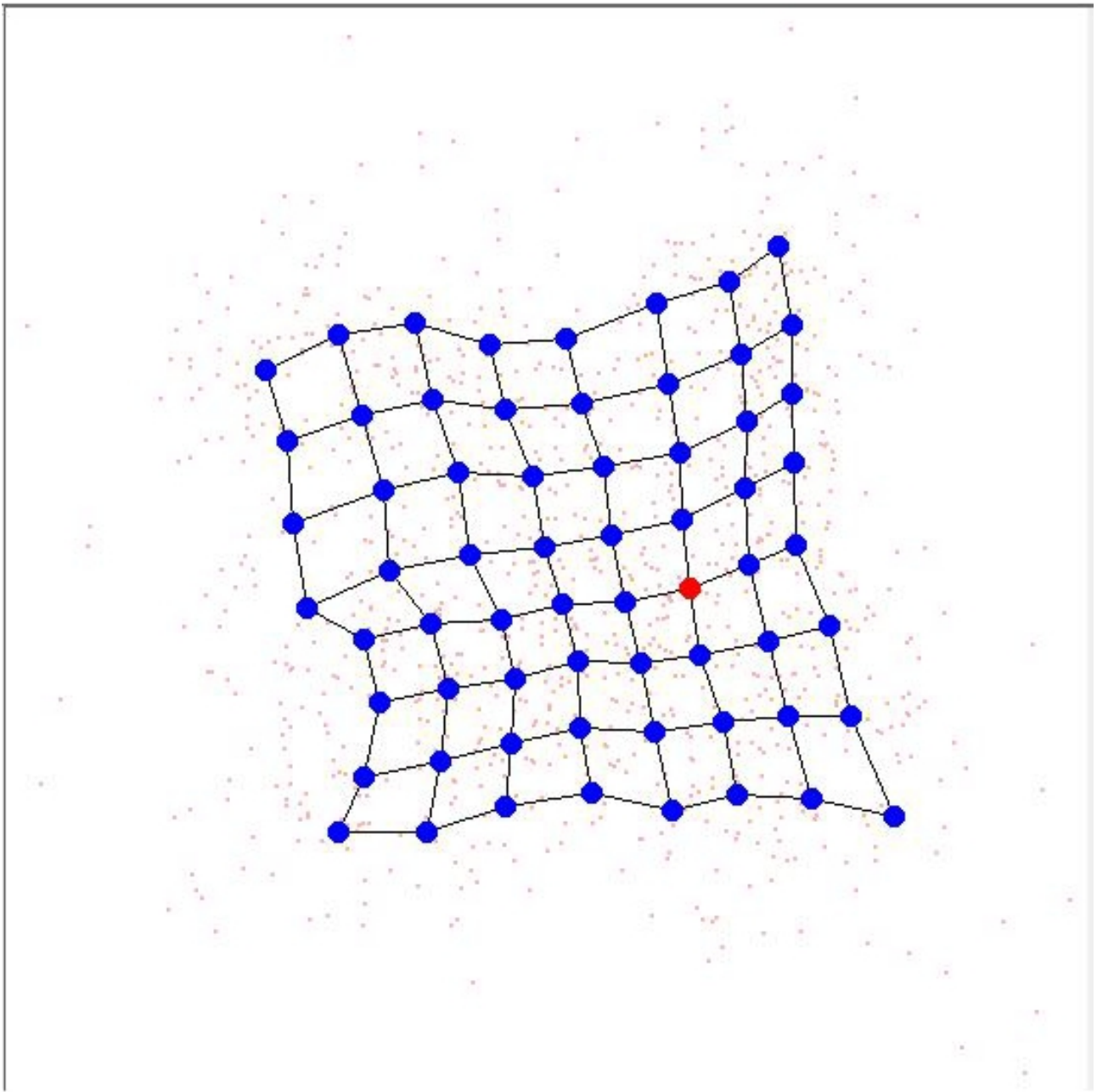
Pour data1, nous utilisons une fonction gaussien 1. Nous voyons clairement ressortir les 5 classes. Il faudra donc placer un point d'initialisation sur chaque nuage de points.

Data 2 :



Pour data2, nous utilisons la fonction winner. On voit très clairement les 3 classes à droite et celle au centre. En ce qui concerne la partie gauche de l'image, on distingue plus difficilement 3 classes. Il faudra donc placer les points d'initialisation en conséquence.

Data 3 :



Pour data3, nous utilisons la fonction gaussian1. Ici nous voyons clairement le problème, la carte est étalée sur l'ensemble des points. Il n'y a pas vraiment de classification possible pour data3.

Étude des données du sonar

Classification

Les données sont représentées sur 60 dimensions, il est donc impossible d'avoir une vision globale. En revanche, il est possible d'analyser statistiquement les données pour se rendre compte que certaines dimensions sont beaucoup plus utiles pour la classification que d'autres.

En utilisant les k-means sur les données du sonar, il est naturel de penser à mettre 2 classes : Mine et Rock. Or on s'aperçoit qu'il va très mal classé, sur un centroïde sera rattaché autant de Mine que de Rock et sur l'autre 72 % de Mine et 28 % de Rock. On peut donc essayer 4 classes : Mine, Rock, peut être Mine(qui contiendrait beaucoup plus de Mine que de Rock), peut être Rock (qui contiendrait beaucoup plus de Rock que de Mine). On va bien avoir 2 classes avec une petite majorité de Mine (70%) mais les 2 autres classes seront répartis équitablement entre Mine et Rock. On pourrait essayer encore d'augmenter le nombre de classes, mais il est clair que d'un point de vu donnée, certains Rock ressemblent à des Mine et vice-versa.

Essayons donc d'utiliser les cartes auto-organisatrice de Kohonen. On lancera l'apprentissage sur une carte 5x5. On se rend compte que celui-ci arrive à bien isoler la plupart des patterns entre eux mais il reste encore une bonne partie du corpus mélangés. Si on essaye avec une carte plus grande, on obtient les mêmes résultats avec des parties de la carte non utilisées. On a donc quelques données, qui se ressemblent mais qui ne sont pas du même type, qui posent problème pour la classification.

Discrimination

Nous avons décidé de diviser le corpus de cette manière : 50 % pour l'apprentissage et 50 % pour les tests.

Après de très nombreux essais de différentes architectures, nous en avons retenus deux qui donnent d'assez bons résultats.

Un perceptron multicouche avec 1 couche cachée : 60(L) x 60(TanH) x 2(G). Nous avons obtenu les matrices de confusion suivantes pour un apprentissage de 500 époques :

Learn Confusion Matrix

	delivred			
desiréd		Mine	Rock	Sum
	Mine	96,49% (55)	03,51% (2)	54,81% (57)
	Rock	06,38% (3)	93,62% (44)	45,19% (47)
	Sum	55,77% (58)	44,23% (46)	95,19% (99 / 104)

Test Confusion Matrix

	delivred			
desiréd		Mine	Rock	Sum
	Mine	81,48% (44)	18,52% (10)	51,92% (54)
	Rock	26,00% (13)	74,00% (37)	48,08% (50)
	Sum	54,81% (57)	45,19% (47)	77,88% (81 / 104)

On voit que le perceptron classe moyennement bien les Mine et les Rock avec un total de 77,88 % de bon classement.

Et un perceptron multicouche avec 2 couches cachées : 60(L) x 10(S) x 10(T) x 2(G). Nous avons obtenu les matrices de confusions suivantes pour un apprentissage de 500 époques :

Learn Confusion Matrix

	delivred			
desiréd		Mine	Rock	Sum
	Mine	98,25% (56)	01,75% (1)	54,81% (57)
	Rock	57,45% (27)	42,55% (20)	45,19% (47)
	Sum	79,81% (83)	20,19% (21)	73,08% (76 / 104)

Test Confusion Matrix

	delivred			
desiréd		Mine	Rock	Sum
	Mine	98,15% (53)	01,85% (1)	51,92% (54)
	Rock	56,00% (28)	44,00% (22)	48,08% (50)
	Sum	77,88% (81)	22,12% (23)	72,12% (75 / 104)

CARON Frédéric
CHARLES Mickaël

Pour celui-ci, nous obtenons des résultats globaux légèrement moins bien. Par contre, il faut remarquer qu'il classe mieux les Mine et moins bien les Rock, chose qui peut se révéler plus efficace dans la vraie vie. Il sera préférable de mal classer un rock plutôt qu'une mine.

Spike-Timing-Dependent Plasticity

Les paramètres utilisés dans l'implémentation sont : C la capacité (de la membrane ?), R la résistance (de la synapse ?), V le seuil de décharge du neurone, $w1$ et $w2$ que l'on a choisis constant mais qui peuvent être des fonctions du temps, $t0$ l'intervalle de temps de prise en compte des spikes pré-synaptique, *duree* le temps d'exposition d'une entrée, dt l'intervalle de temps entre deux mises à jour des neurones et *epoch* le nombre de fois qu'on présente l'ensemble du corpus d'apprentissage aux neurones. La partie de test n'a pas été implémentée car nous n'avons pas trouvé les bons paramètres pour obtenir un apprentissage viable (sachant les résultats obtenus sur Ginnet). Il semblerait que l'excitation latérale soit trop important à côté de celle des entrées, poussant ainsi les neurones à spiker sans prendre en compte une quelconque étiquette.

Classification de signaux de parole

Pour effectuer la classification, nous avons divisé le corpus comme suit : 67 % pour l'apprentissage et 33 % pour les tests. Après avoir testé plusieurs architectures, les meilleurs résultats ont été obtenu avec un perceptron sans couche cachée : 240(L) x 39(G). Nous obtenons 83 % de réussite en apprentissage et 71 % en test. Il est très difficile d'obtenir de meilleurs résultats car le nombre d'entrées/dimensions est trop élevés. D'ailleurs on peut remarquer sur la matrice de confusion, que certains phonèmes n'arrivent pas du tout à être bien classés.