ML - TME 3

CHERCHOUR Lièce & DIEZ Marie

Avec l'aide de THAUVIN Dao & STERKERS Luc

1 Expérimentations

1.1 Descente de gradient sur le problème à deux gaussiennes

Nous allons observé l'évolution de la frontière de decision à l'aide d'une descente de gradient avec la régression linéaire et une avec la régression logistique.

Nous allons initialisé notre W représentant la droite de décision à [1,-1], cela nous permettra de plus facilement observer l'évolution de la frontière à travers les itérations.

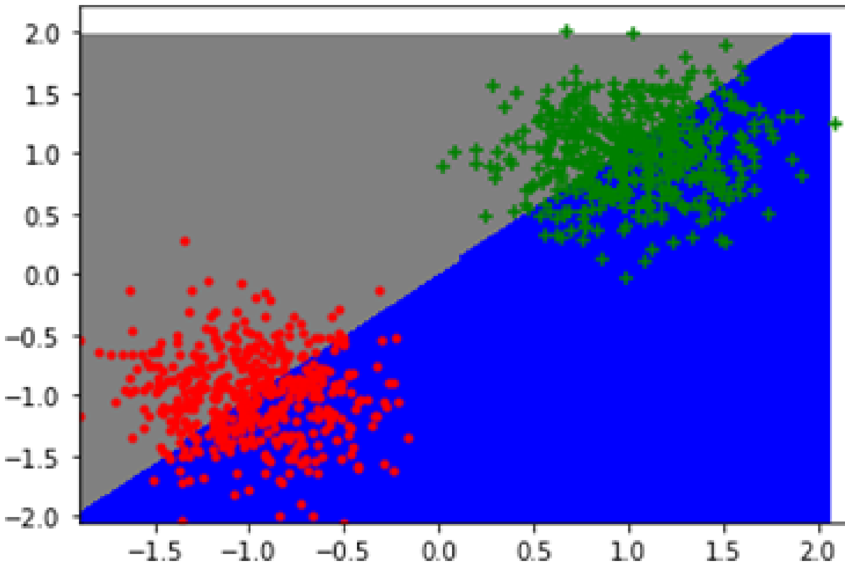


Figure 1: Frontière de décision avec w = [1,-1]

On effectuera sur chacun d'entre elle 1000 itération à pas de 250, pour un epsilon de $10e-3$

1.1.1 Résultat avec la régression linéaire

.

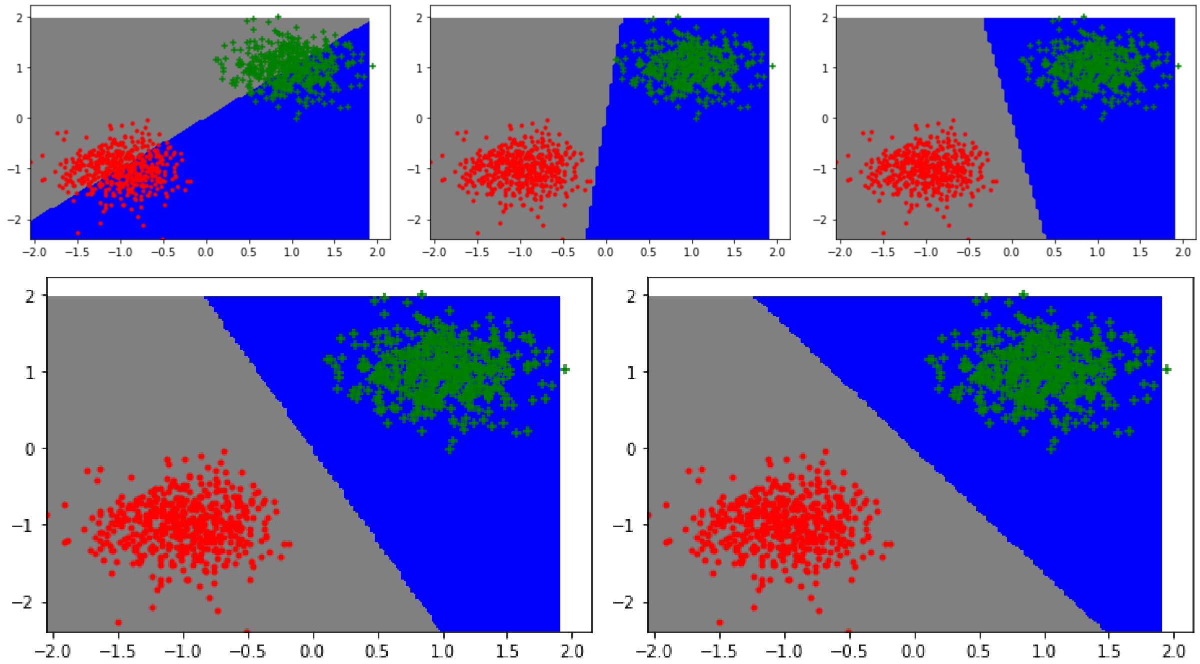


Figure 2: Evolution de la frontière de décision avec la régréssion linéaire

1.1.2 Résultat avec la régression logistique

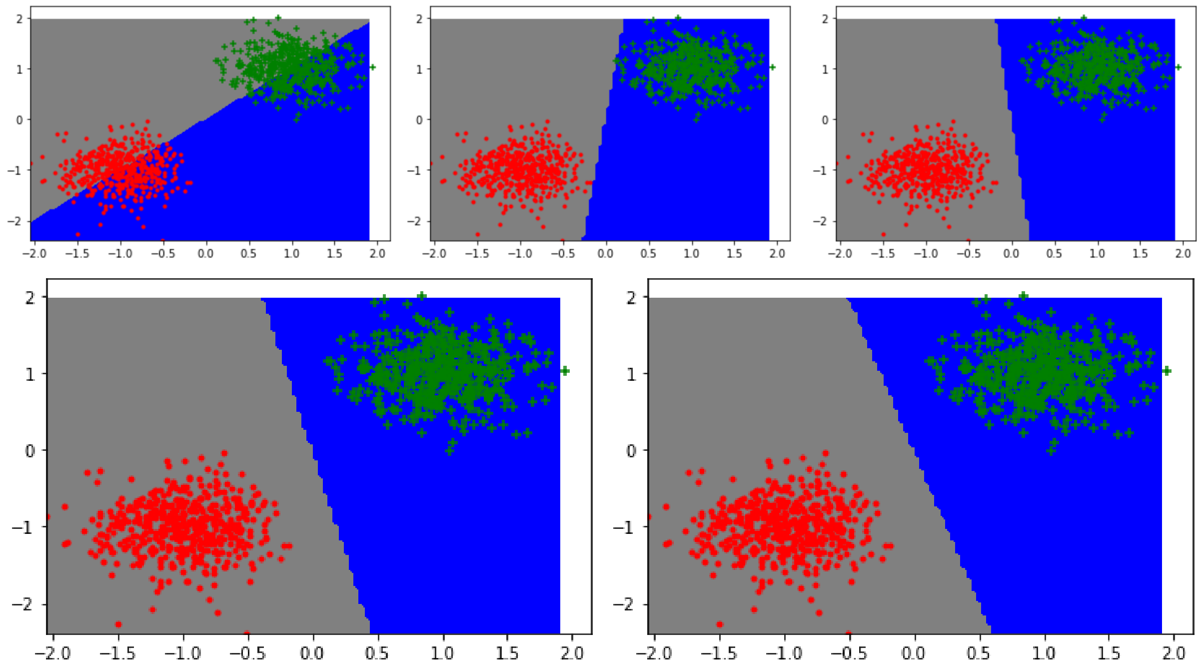


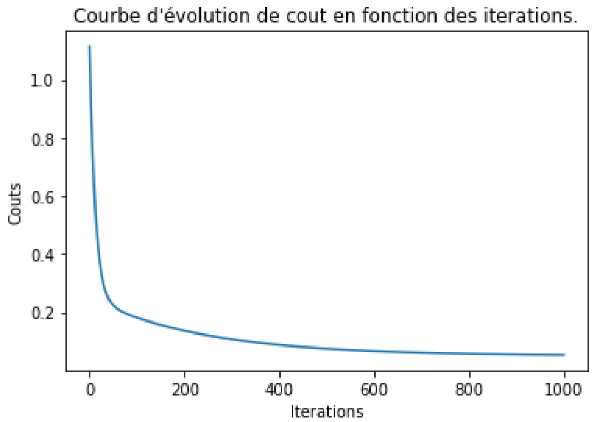
Figure 3: Evolution de la frontière de décision avec la régréssion logistique

1.1.3 Observations

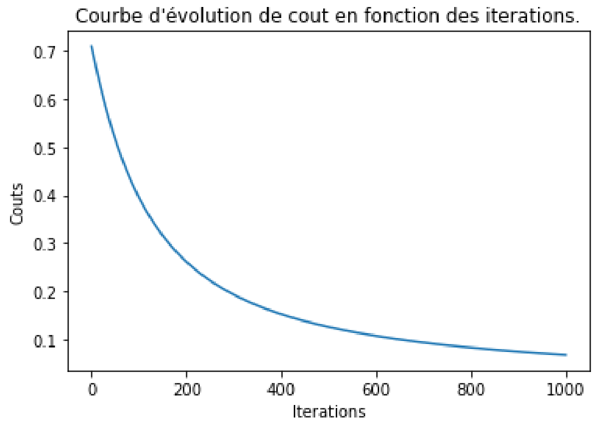
On peut facilement voir que les deux fonctions de gradient permet d'obtenir un résultat semblable, la frontière de décision est un peu différente. La régression logistique possède une frontière de décision plus proche de ces données, dans le cas de valeur aberrantes cela risque de poser plus de problème qu'en régression linéaire.

Voici les graphiques de l'évolution du coût en fonction des itérations :

(a) Régression linéaire



(b) Régression logistique



On peut voir que la régréssion linéaire arrive à réduire rapidement son coût mais prends un peu plus de temps à se stabilisé. À contrario, la régréssion logistique effectue toujours le même pas peut importe le nombre d'itérations.

1.2 Influence du pas du gradient sur la convergence

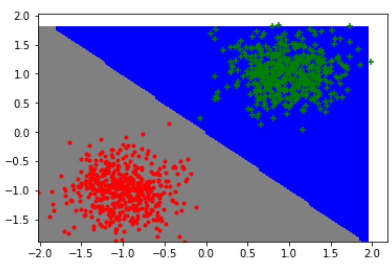
Notre pas initiale était de $\epsilon = 10e - 3$

Nous allons maintenant observer les résultats en augmentant, puis en réduisant drastiquement l'échelle de notre pas. Pour l'augmentation, nous utiliserons un pas de $\epsilon = 0.2$, pour la réduction, nous utiliserons un pas de $\epsilon = 10e - 6$

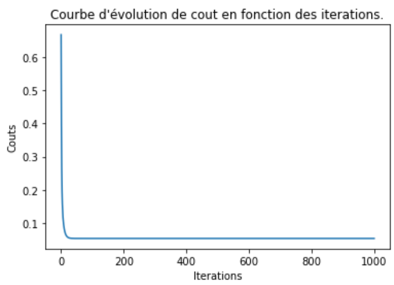
Pour montrer la direction de notre algorithme d'optimisation, nous allons également affiché sa trajectoire dans le graphe de la fonction de coût dans l'espace des poids.

1.2.1 Résultat avec la régression linéaire pour un pas rapide

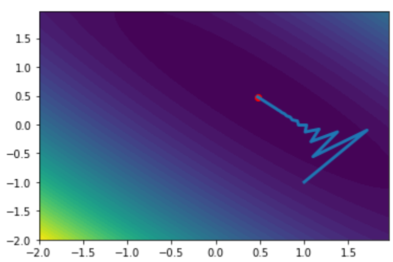
(a) Frontière de décision



(b) Evolution du coût

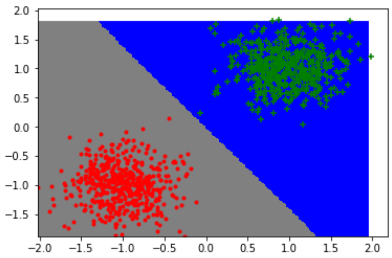


(c) Trajectoire

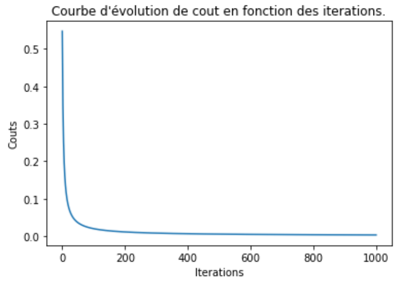


1.2.2 Résultat avec la régression logistique pour un pas rapide

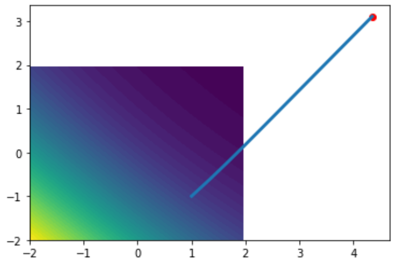
(a) Frontière de décision



(b) Evolution du coût

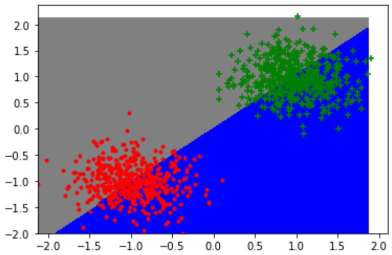


(c) Trajectoire

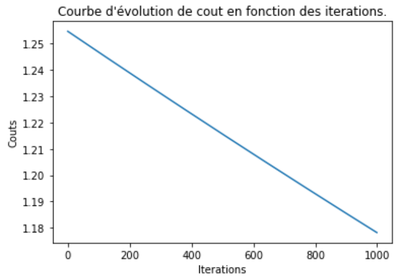


1.2.3 Résultat avec la régression linéaire pour un pas lent

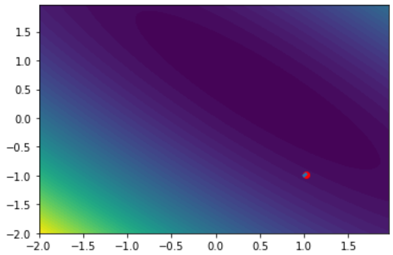
(a) Frontière de décision



(b) Evolution du coût

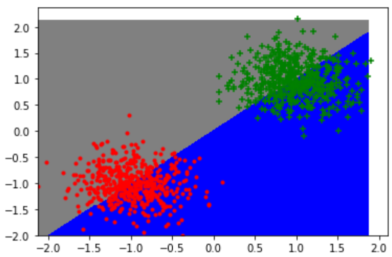


(c) Trajectoire

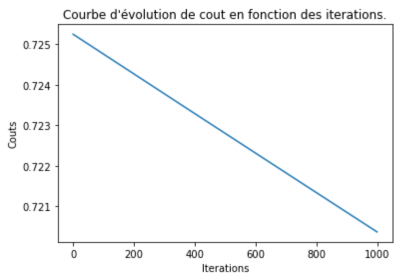


1.2.4 Résultat avec la régression logistique pour un pas lent

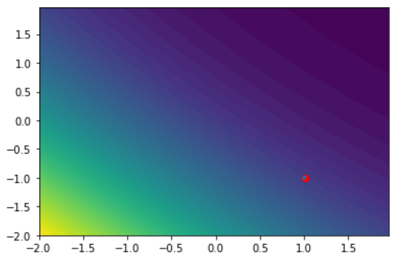
(a) Frontière de décision



(b) Evolution du coût

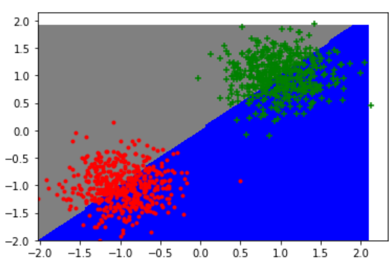


(c) Trajectoire

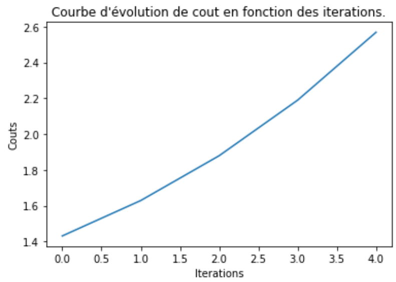


1.2.5 Résultat avec la régression linéaire pour un pas qui devrait faire divergé

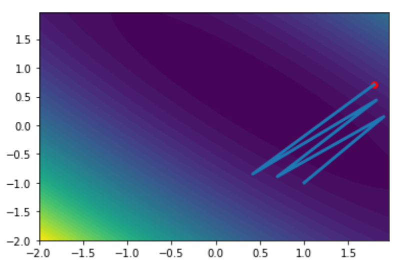
(a) Frontière de décision



(b) Evolution du coût

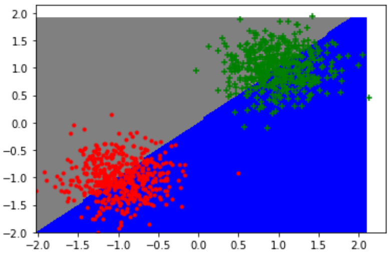


(c) Trajectoire

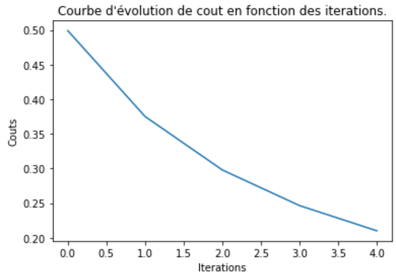


1.2.6 Résultat avec la régression logistique pour un pas qui devrait faire divergé

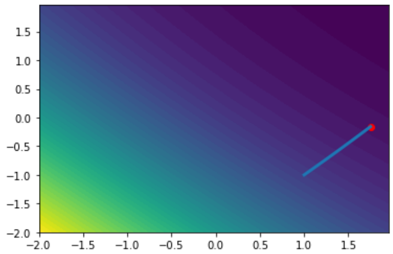
(a) Frontière de décision



(b) Evolution du coût



(c) Trajectoire



1.2.7 Observations

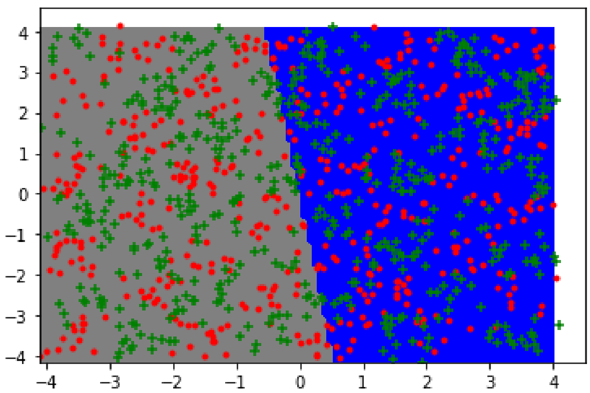
On remarque tous d'abord que la sensibilité au pas de ces deux fonctions sont assez similaires sauf dans le cas limite. Effectivement on observe que pour des pas ϵ très grand ou l'on devrait normalement observer une divergence, ce n'est pas le cas pour l'algorithme de régression logistique. Effectivement, on observe que meme avec un pas énorme, même si il prend plus de temps pour atteindre son minima comme le montre notre fonction de coût. On peut également regarder du côté des trajectoires emprunté par nos algorithmes à différents pas, MSE rebondit et ne s'éloigne de plus en plus de son minimum, dans le cas de la régréssion logistique, il ne fait que s'approcher du minimum comme on peut le voir dans la 1.2.6.

1.3 Problème non séparable

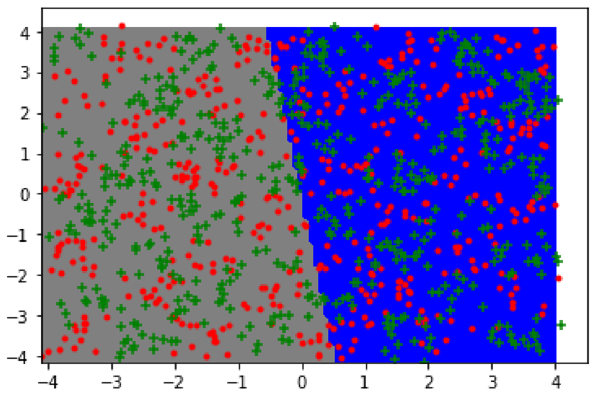
Nous avons jusqu'ici seulement travailler sur des problèmes séparables, nous allons maintenant observer notre algorithme sur un problème non séparables, comme l'échiquier par exemple.

Nous allons générer des données sous formes d'échiquiers à l'aide de la fonction *gen_arti*, nous avons obtenu ces frontières de décisions après 1000 itérations :

(a) Régression linéaire



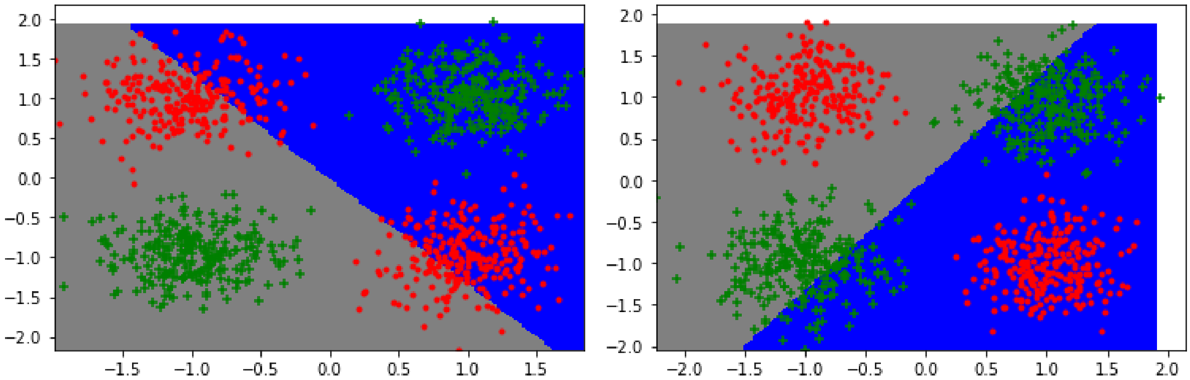
(b) Régression logistique



Si nous générons des données sous le format 4 gaussiennes, nous obtenons :

(a) Régression linéaire

(b) Régression logistique



On peut observer que notre algorithme ne s'en sort pas et n'arrive pas à trouver une frontière de décision qui pourrait répondre à ce problème, c'est tout à fait normal.

Pour certains problèmes non séparables linéairement, l'ajout de dimension peut la rendre linéairement séparables. Ici, cela ne marcherait pas.