COMPTE RENDU

MACHINE LEARNING

TP 2 : modèles linéaires pour la classification

Enseignant : HIDANE Moncef
Étudiant : LE Quang Linh & PHAM Ba Dat

GSI-5A-ACAD (2020-2021)

Blois, 06/11/2020

I. INTRODUCTION

Ce TP de Machine Learning concerne les modèles linéaires pour la classification linéaire. Il comporte 2 parties : données simulées et données réelles. Les essais seront étudiés par les jeux des données simulées pour trouver les résultats des méthodes. Et les données réelles sont utilisées pour prédire l'espèce des Iris, par le taux de bonne classification.

II. DONNEES SIMULEES

1. ***Visualisation des données :***

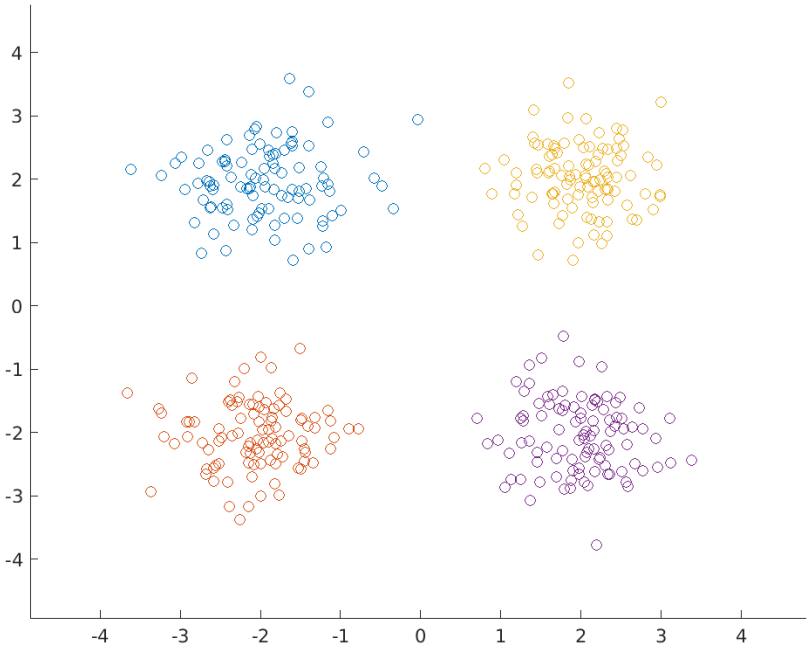

Figure 1 : Visualisation des données

La distribution de ces points de données nous permet de déterminer qu'ils sont générés aléatoires suivi la loi normale avec des valeurs de std et sigma différentes.

2. ***Les conventions :***

Si N es le nombre des points de données
M est le nombre des features
K est le nombre de classes

Nous avons donc la taille des matrices comme suivant :
X : N x M
T : N x 1
ToneofK : N x K
W : M x K

3. ***Visualisation l'évolution des itérations du descente de gradient :***

Visualisation pour le cas : K = 4, itérations = 1000, alpha = 0.1

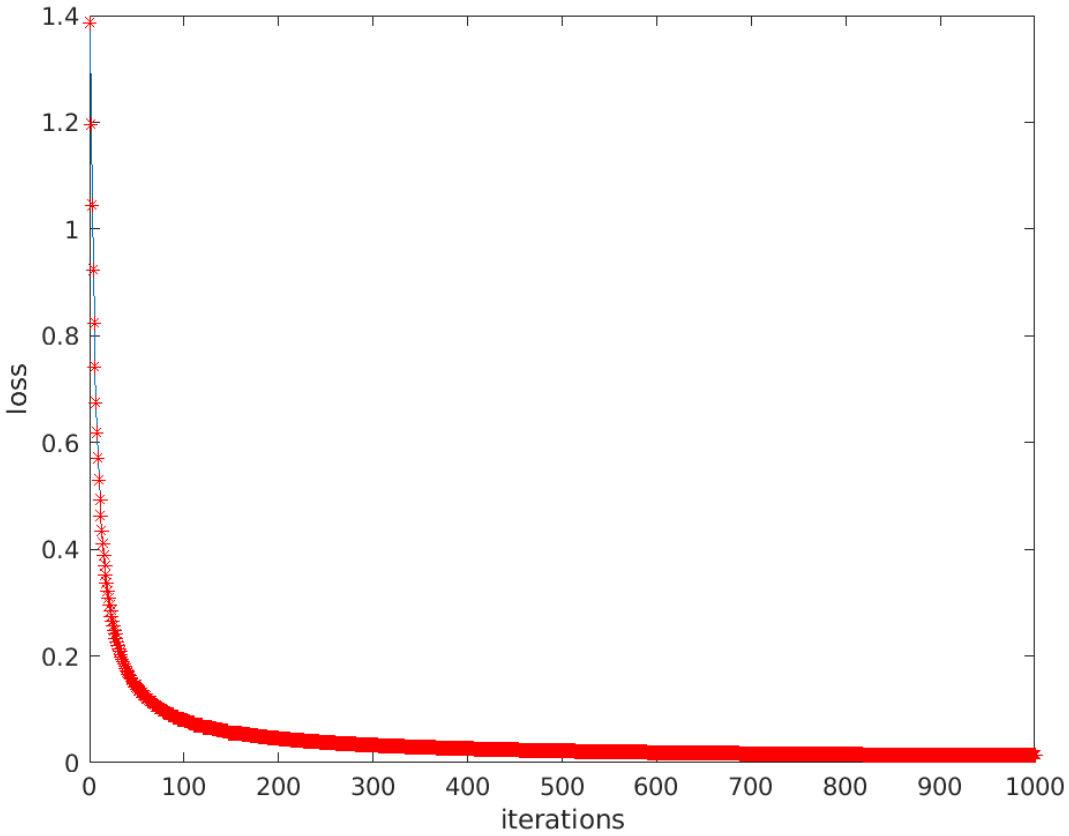

Figure 2 : Descente de gradient

Cette figure nous montrer que notre modèle est bien appris. La valeur de l'erreur est diminuée lorsque le numéro d'intégration augmente.

4. ***Hyperplan séparateur entre deux classes :***

Au l'hyperplan séparateur, le modèle va donner le même valeur pour les deux classes concernées. Alors, pour trouver l'hyperplan séparateur nous devons résoudre l'équation suivant (prenant deux classes 1 et 2 par exemple) :

$$Xp*W1 = Xp*W2$$

Avec W1 et W2 sont des colonnes du W (W= ([W1 W2 W3 W4])
Xp est un point avec des valeurs d'abscisse et ordonnée correspondant (Xp = [1 x1 x2])

On a donc:

$$W(1,1) - W(1,2) + (W(2,1) - W(2,2)) *x1 + (W(3,1) - W(3,2))*x2 = 0$$

En appliquant la technique ci-dessus pour trouver l'hyperplan séparateur, nous obtenons deux figures de deux cas, avant et après ajouter à l'ensemble d'apprentissage 100 points de la classe 1 autour du point de coordonnées (0, 5) :

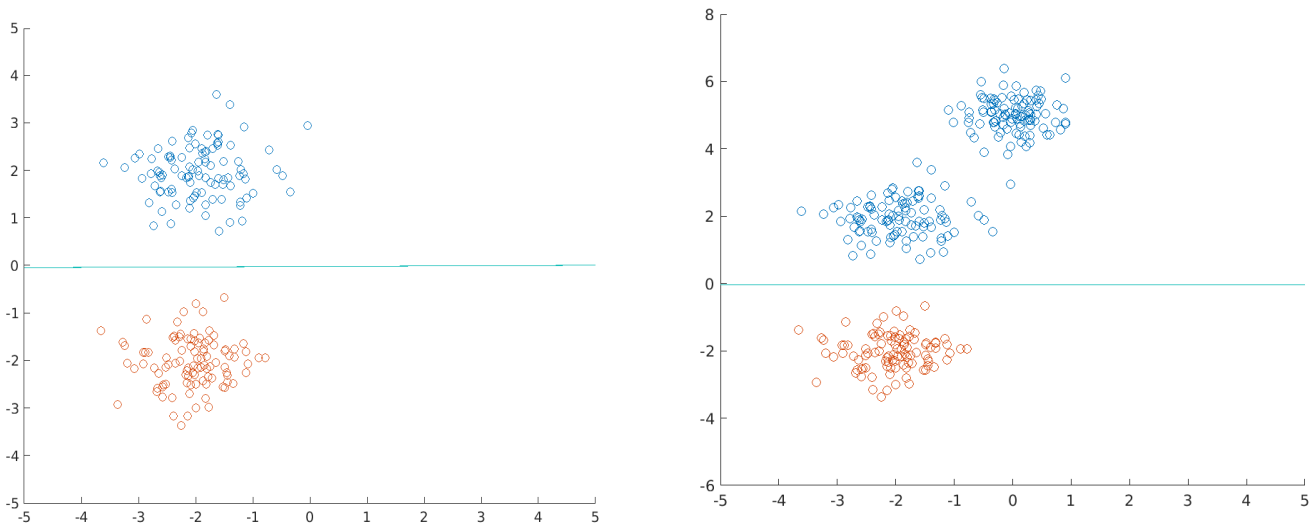

Figure 3 : Hyperplan séparateur entre deux classes 1 & 2

On voit bien que l'hyperplan séparateur ne change pas beaucoup (il y a peut-être un changement mais on ne peut pas voir, ce problème va être résolu dans la partie suivant en utilisant la région de décision) malgré qu'on a ajouté en plus 100 points de la classe 1. C'est parce que le modèle est appris en minimisant la proportion de mauvaise classification mais pas le moindre carrée. Si on utilise la méthode du moindre carré, le changement sera plus clair.

On va tester le résultat pour une classification multiclasse avec K = 3 (on prendra les classes 1, 2 et 3), les hyperplans séparateurs sont suivants :

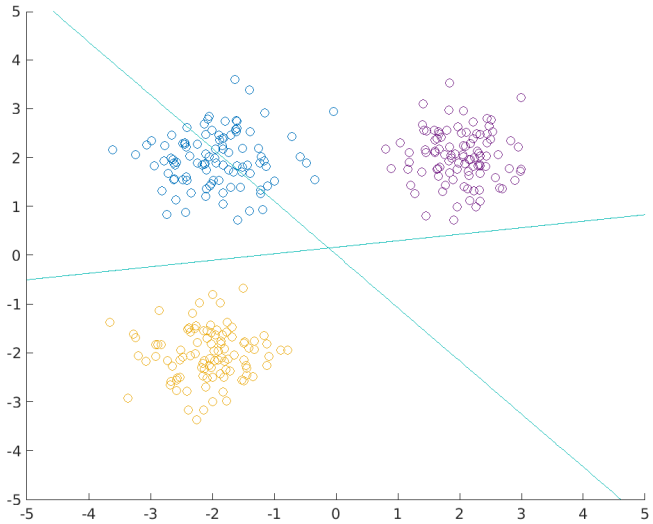

Figure 4 : Hyperplan séparateur pour 3 classes

Pour ce cas, le résultat n'est plus satisfait car la fonction *fcontour* ne marce que pour le cas de 2 classes, pour 3 classes elle affiche des droites sans respecter des groupes de points qui ne sont pas concernés. C'est encore un problème de cette méthode !

5. ***Région de décision :***

Pour résoudre des problèmes de l'hyperplan séparateur, on va utiliser la méthode de région de décision.

La classification multiclasse avec K = 4 pour deux cas : avant et après ajouter à l'ensemble d'apprentissage 100 points de la classe 1 autour du point de coordonnées (0, 5) :

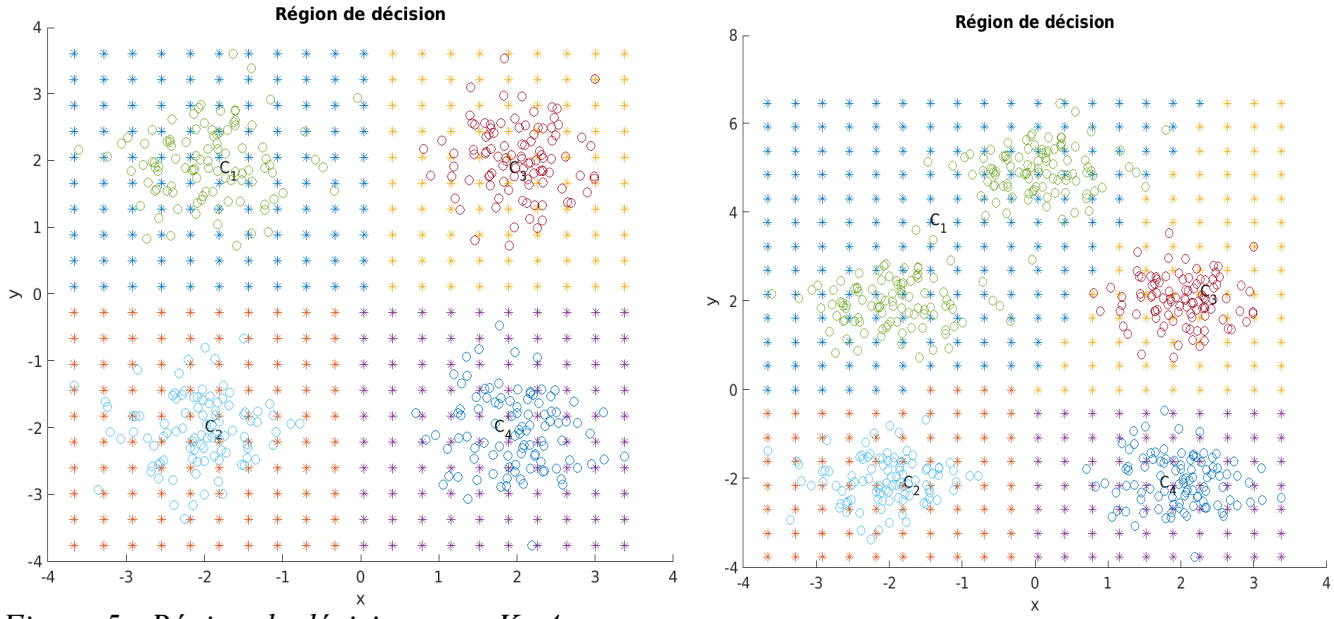

Figure 5 : Région de décision pour K=4

Voilà, pour la visualisation, cette méthode est plus lisible et claire comparaissant avec l'hyperplan séparateur. En affichant l'hyperplan séparateur on ne peut pas quand même voir le changement.

6. ***Des différences d'importances entre les erreurs de classification :***

Si on prend la matrice L comme suivant, on doit reprendre la figure 5 (figure à gauche) :

```
L0 = [0, 1, 1, 1;...
      1, 0, 1, 1;...
      1, 1, 0, 1;...
      1, 1, 1, 0];
```

Si on prend la matrice L comme suivant, la classe 1 sera plus toléré que la classe 3 (la décision du modèle va etre plus facile avec la classe 1 mais difficile avec la classe 3) :

```
L1 = [0, 1, 5, 1;...
      1, 0, 1, 1;...
      1, 1, 0, 1;...
      1, 1, 1, 0];
```

Si L est comme suivant, la classe 1 sera plus toléré que 3 et la classe 4 sera plus toléré que 2 :

```
L2 = [0, 1, 5, 1;...
      1, 0, 1, 1;...
      1, 1, 0, 1;...
      1, 5, 1, 0];
```

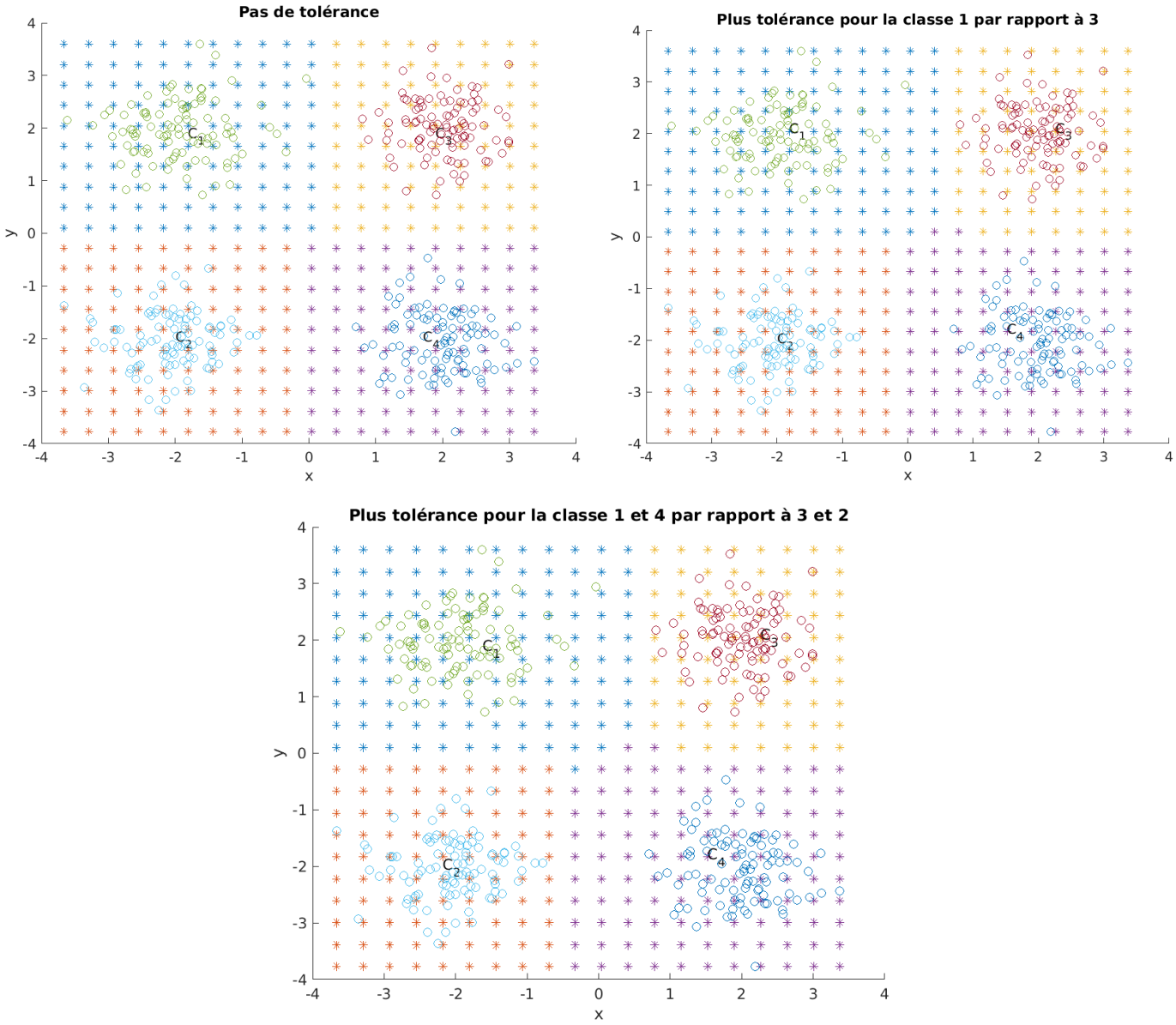

Figure 6 : Région de décision avec des différence L

Cette technique est souvent utilisée lorsqu'on veut faire attention à une classe spécifique. Par exemple, pour le cas du classe 1 et 3 ci-dessus, on veut que le modèle va être difficile à prendre la décision pour un point qui est dans la classe 3 mais plus facile pour qu'il va être dans la classe 1. C'est-à-dire, on peut accepter la fausse classification pour la classe 1 mais pour la classe 3, on ne veut pas. Ça veut dire que si on affiche la probabilité d'un point qui est prédit pour la classe 3, cette valeur va très proche à 1 (pour la classe 1, ce n'est pas sûr que les probabilités soient proches à 1).

7. ***Régulation de la régression logistique :***

Approche non-régulé peut nous donne des paramètres avec des valeurs très grands. On va donc résoudre ce problème en ajoutant un terme pour contrôler l'amplitude des paramètres. Cette technique est appelée la régulation. De plus, la régulation est utilisée pour résoudre le problème d'overfitting.

La figure suivante compare les paramètres W sans et avec la régulation :

W × W_reg ×

3×4 double

	1	2	3	4
1	1.0847	0.9826	0.9148	1.0178
2	-0.6587	-0.6366	2.6681	2.6272
3	2.7409	-0.5981	2.5192	-0.6620

W × W_reg ×

3×4 double

	1	2	3	4
1	1.0842	0.9823	0.9163	1.0172
2	-0.6848	-0.6668	2.4615	2.4200
3	2.5218	-0.6297	2.3263	-0.6885

Figure 7 : Les paramètres W sans et avec régulation

La régulation n'affecte pas beaucoup dans notre cas. C'est parce que des groupes de points d'apprentissage sont bien séparés (loin l'un à l'autre). La deuxième vient des caractéristiques des données, niveau d'importance des caractéristiques est pareil. Par contre, on voit encore des paramètres avec des valeurs plus de 2 qui diminuent un peu.

La figure suivante montre que la descente de gradient pour le cas non-régulé est un peu plus vite comparaissant avec le cas de la régulation (risque de problème d'overfitting lors de training mais ici on ne compte pas ce phénomène) :

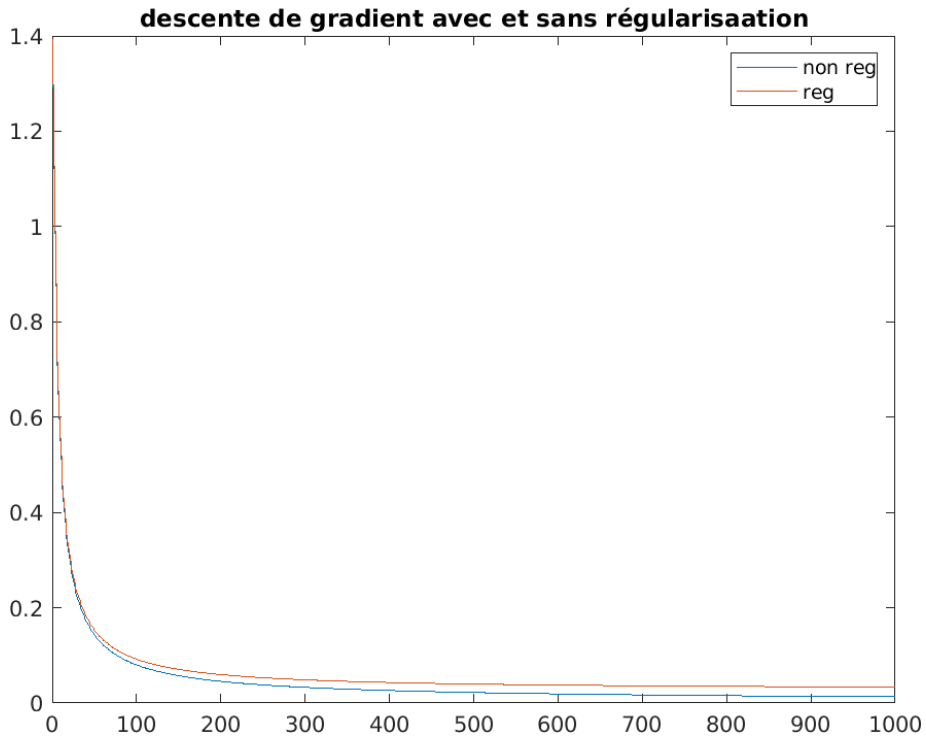

Figure 8 : Descente de gradient avec et sans régulation

On voit bien qu'avec la régulation, le temps pour que l'erreur s'annule est un peu plus. C'est expliqué par le fait que les paramètres W est supposée aux contraintes dans ce cas. Il est donc plus difficile pour qu'elle s'ajuste.

III. DONNEES REELLES :

Distribution des données d'apprentissages

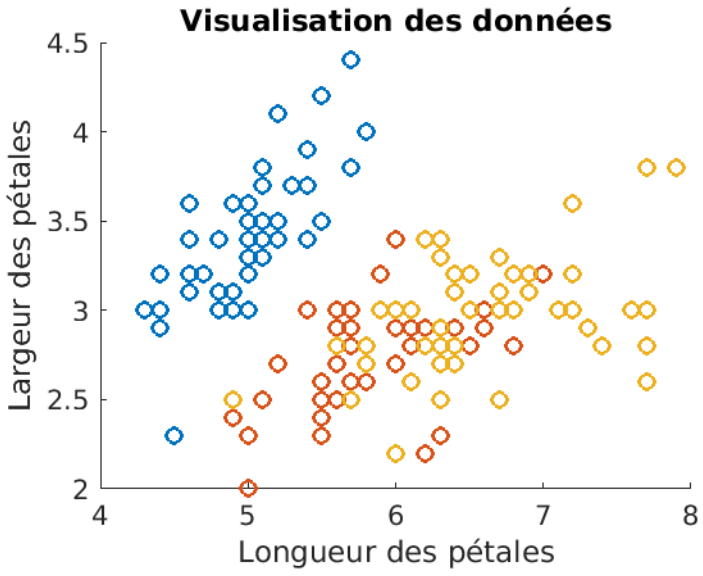

Figure 9 : Distribution des données d'apprentissages

Approche 1 : modèle simple avec des données brutes

Pour ce cas, on utilise des données brutes dont 2 caractéristique2 : longueur et largeur des pétales. On va prendre la méthode de régression logistique avec la régulation. Des hyperparamètres sont :

n_iters=10000, alpha=0.01, gamma=0.5

Après training on a pris des figures de learning curve suivant :

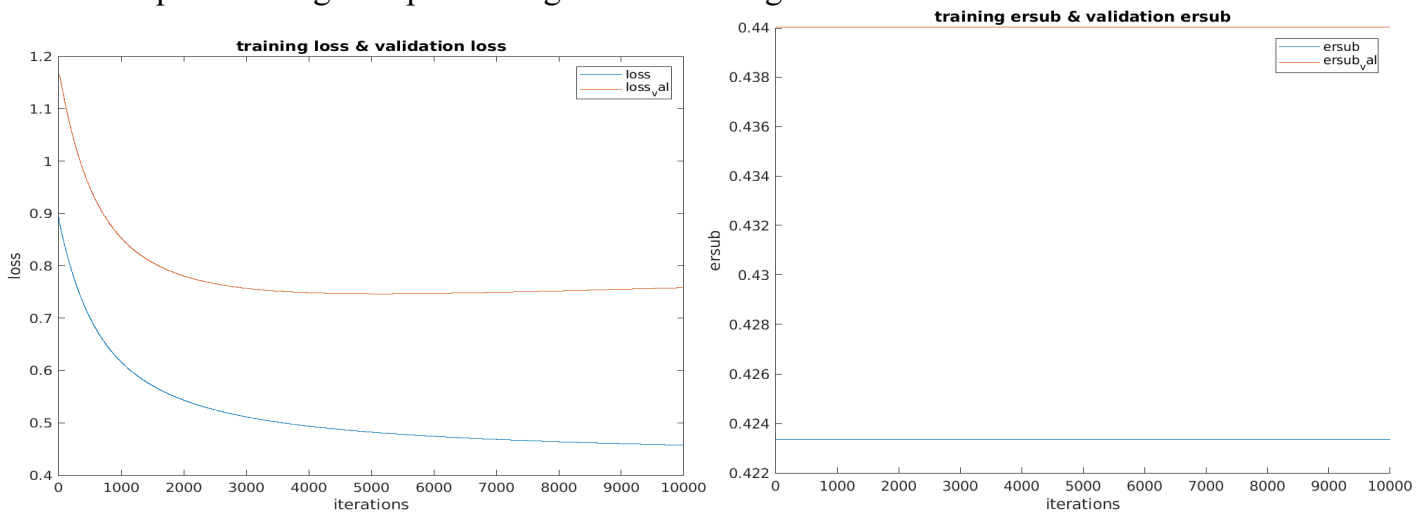

Figure10 : Learning curve de l'approche 1

On voit bien que l'évolution de loss_val est commencé à augmenter après 4000 itérations (Si on utilise early stopping, les meilleurs paramètres peut être trouvé après presque 4000 itérations, mais même si on fait ça le modèle est encore mal). De plus, le gap entre loss et loss_val est très grand aussi nous dit que le modèle est 'high bias'. En outre, les valeurs de ersub et ersub_val sont toujours mal nous dit que le modèle est échoué.

Le nuage des points peut expliquer ce résultat. On voit bien que ce n'est pas possible de bien séparer les trois types de fleurs. Les classe 2 (versicolor) et 3 (virginica) se fusent l'une dans l'autre.

Des régions de décision :

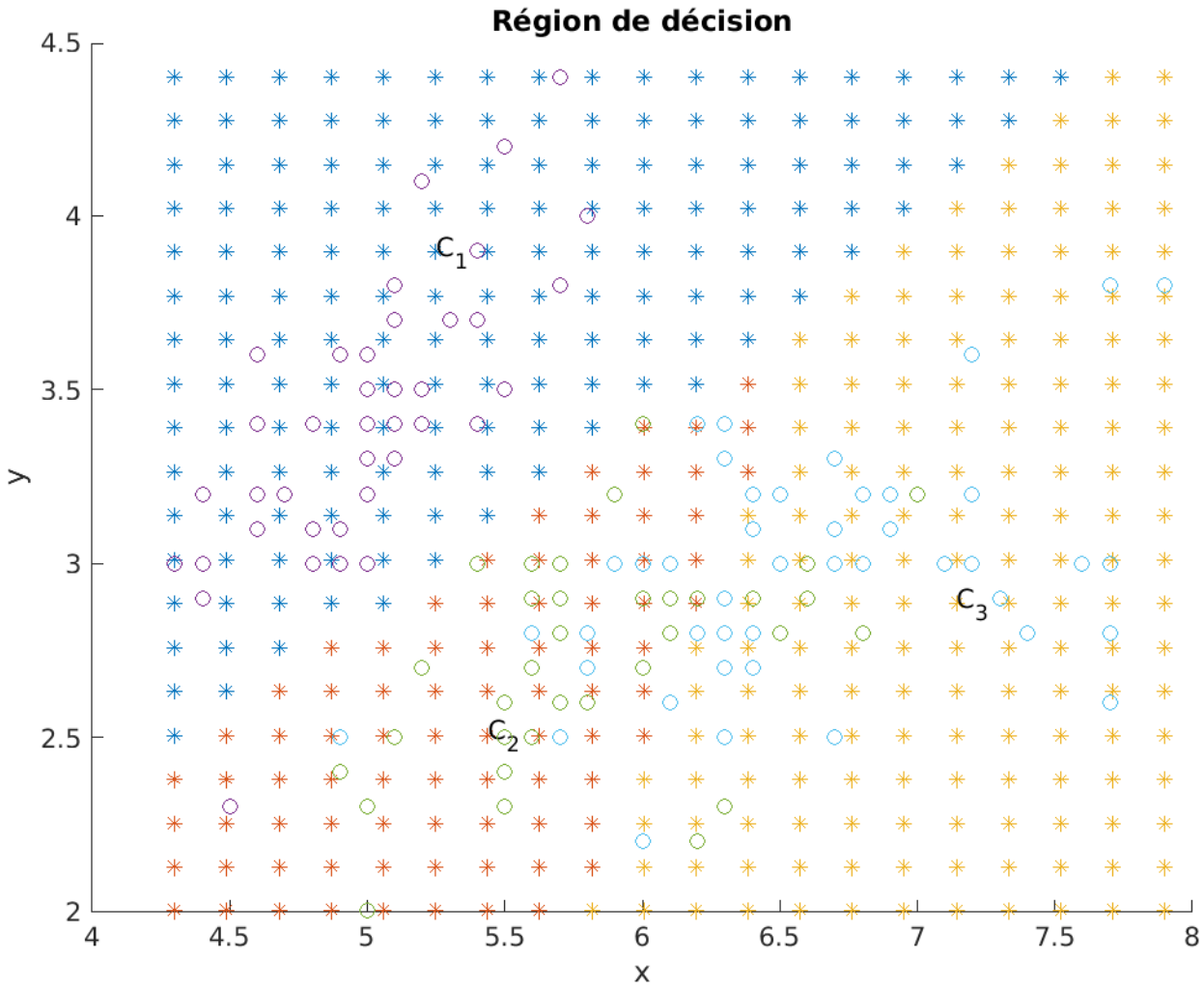

Figure11 : région de décision de l'approche 1

La moyenne de l'erreur de classification sur les 10 expérimentes avec ce modèle est :

ersub_10 × ersub_val_10 ×

1×1 double

	1
1	0.4233

ersub_10 × ersub_val_10 ×

1×1 double

	1
1	0.4400

Pour résoudre ce problème, il faut qu'on pense à utiliser l'approche 2 qui va augmenter la dimension des caractéristiques des données d'apprentissages et aussi les transformer par une fonction spécifique.

Approche 2 : Utilisation la transformation sinusoïdale des données brutes

On va modifier des données d'apprentissage comme suivant :

X= [sinx1, sinx2, cosx1, cosx2]

Après avoir testé plusieurs valeurs pour des hyperparamètres, nous avons trouvé des valeurs acceptables suivant :

n_iters=10000, alpha=0.02, gamma=1

Le learning curve qu'on a trouvé :

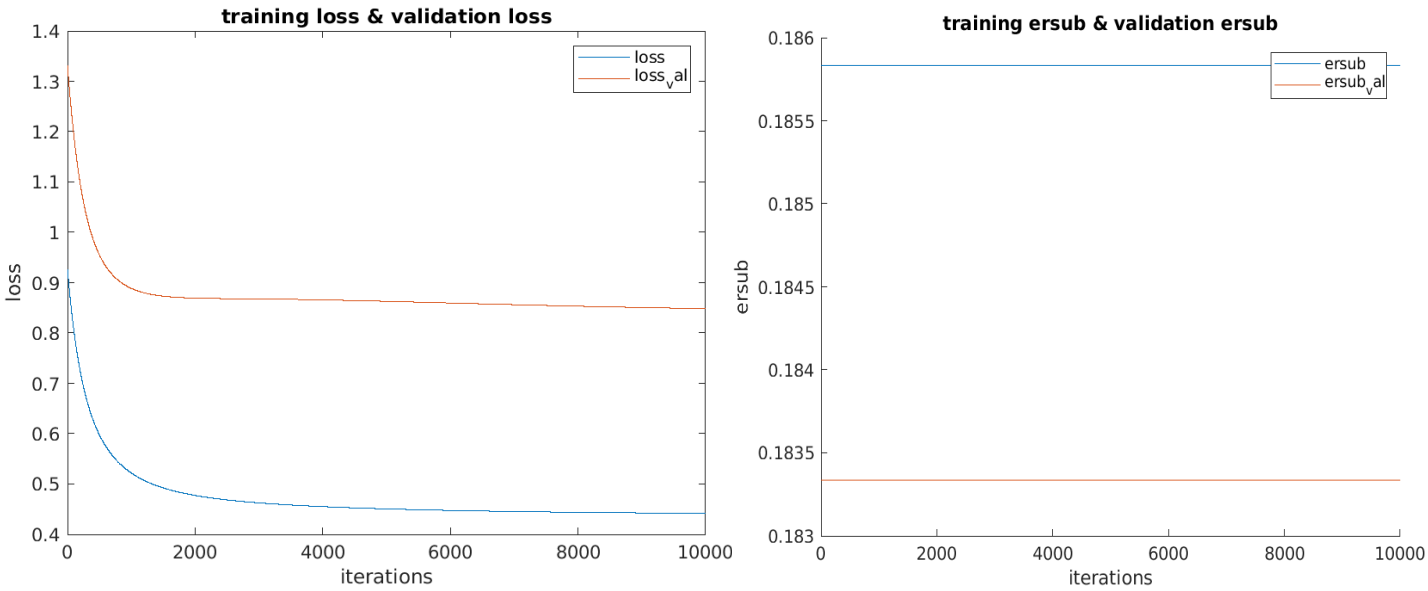

Figure12 : Learning curve de l'approche 2

La figure 12 montre une grande amélioration de la régression. Les valeurs de ersub et ersub_val diminuent de manier clairement. Ces deux valeurs sont très proches nous dit que notre modèle est déjà pas mal. La moyenne de l'erreur de classification sur les 10 expérimentes avec ce modèle est :

ersub_10 | ersub_val_10

1×1 double

	1
1	0.1858

ersub_10 | ersub_val_10

1×1 double

	1
1	0.1833

Des régions de décision :

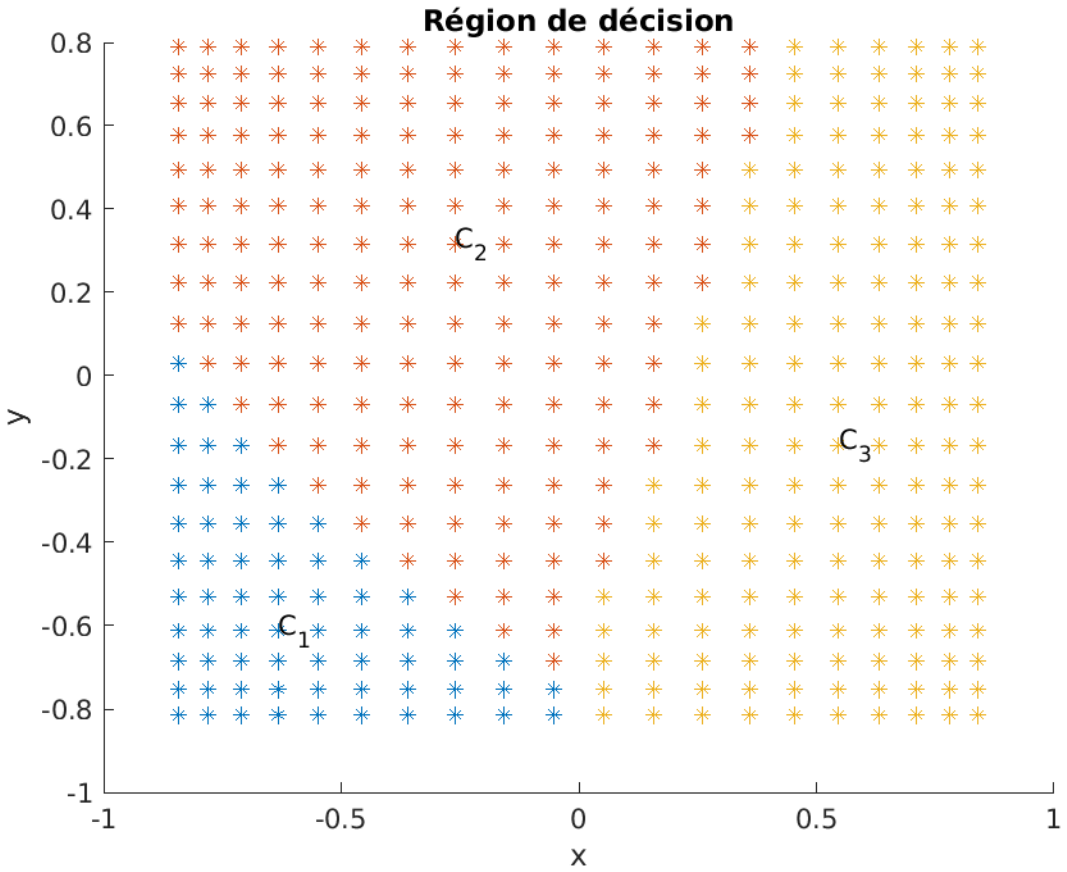

Figure13 : région de décision de l'approche 2

On ne peut pas encore visualiser des données d'apprentissages car ils sont déjà en 4 dimensions mais pas 2. Malgré ça, grâce à la figure 13, on peut croire que les 3 classes sont aussi beaucoup mieux séparées.

IV. CONCLUSION

Ce TP nous donne une perspicacité pour la classification logistique, une méthode très importante à comprendre pour maitriser le domaine de Machine Learning. Nous avons pris connaissance deux méthodes très intéressants pour la visualisation de la séparation des données : l'hyperplan séparant et région de décision. De plus, nous avons étudié deux méthodes de la classification : classification logistique sans et avec la régulation. Nous avons donc vu la différence entre deux méthodes et l'effet de la régulation sur des coefficients. En fin, nous avons travaillé avec des données réelles, vu le problème de la distribution des données et la dimension des données (ici 2 et 4) qui ont un grand l'effet sur le résultat de la classification.