TP Optimisation : Les SVM Pour la classification

AITBAALI Hamza, AMINI Nada, ESSAYEGH Nour,LEFDALI Rida

November 2020

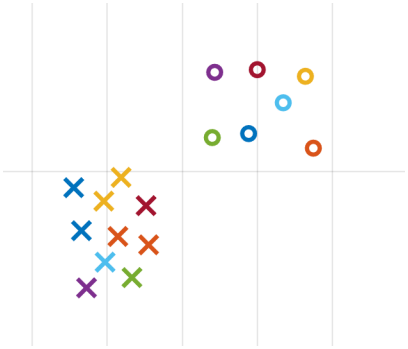
1 Introduction :

Les SVM sont des méthodes d'apprentissage supervisé destinées à résoudre des problèmes de classification. Ce sont des méthodes utilisées dans plusieurs domaines notamment en bio-informatique.

Dans ce TP nous allons modéliser un problème de classification de cellules, saines et cancéreuses. En premier temps nous disposons d'un ensemble de cellules labellisées en 1 pour les cellules cancéreuses et -1 pour les cellules saines. Ces points sont les points d'apprentissage. Et le but est de construire une hypersurface qui sépare d'une façon idéale les cellules cancéreuses des cellules saines. Pour cela nous allons explorer plusieurs variations du modèle pour l'adapter et le généraliser à des distributions de cellules non séparables linéairement ou avec des outliers. Enfin, nous allons ajouter une cellule supplémentaire pour tester l'efficacité de notre programme.

2 Cas linéaire :

Dans ce premier cas que nous allons traiter, nous allons considérer un ensemble de points qui se divise en deux familles.Les points de chacune sont marqués par un label, soit 1 soit -1.



Mais pour séparer les familles il faut trouver un hyper plan qui satisfait l'équation ${}^t wx+b=0$ mais pour assurer une marge maximale il a fallu ajouter une condition supplémentaire. Le tout conduit à un problème de minimisation du lagrangien qui s'écrit :

$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 + \sum_{k=1}^{p} \alpha_k(1 - l_k({}^t wx_k + b))$

le calcul du gradient nous conduit à un problème dual :

$$\begin{cases} Max_{\alpha \geq 0}(H(\alpha)) \\ \sum_{k=1}^{p} \alpha_k l_k = 0 \\ \alpha_k \geq 0 \end{cases}$$

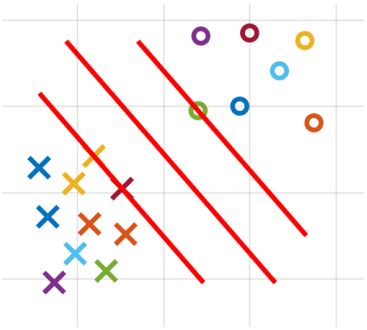
avec $H(\alpha) = -\frac{1}{2}\ {}^t\alpha A\alpha + \ {}^t u\alpha$

Le maximum de la fonction H est déterminé par la méthode du gradient à pas constant et les deux autres conditions consistent à faire une première projection sur l'hyperplan ($\sum_{k=1}^{p} \alpha_k l_k = 0$) puis une deuxième sur le quadrant positif.

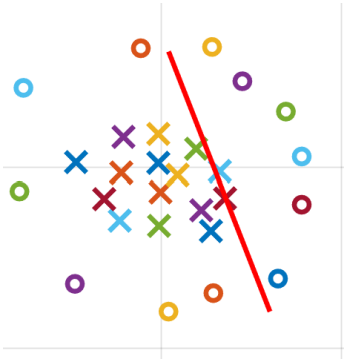
On implémente ces 3 conditions dans une boucle comme suit.

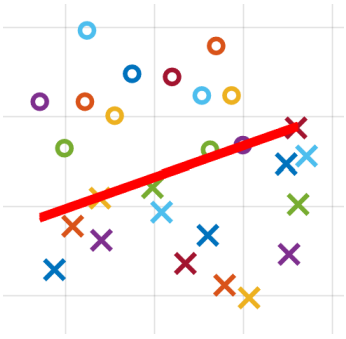
```
% UZAWA
grad_H = -A * alpha0 + u;
alpha_k = alpha0;
vit_conv = zeros(1,npas_max);

while crit_arret >= epsi && npas <= npas_max
    %gradient à pas constant
    alpha_k_plus_1 = alpha_k + pasgrad * grad_H;
    %première projection
    alpha_k_plus_1 = alpha_k_plus_1 - (lab' * alpha_k_plus_1) * lab / (norm(lab)^2);
   %deuxieme projection
    alpha_k_plus_1= max(alpha_k_plus_1,0);
    %recalcul du gradient
    grad_H = -A * alpha_k_plus_1 + u;
    crit_arret = norm(alpha_k_plus_1 - alpha_k);
    vit_conv(npas) = crit_arret;
    alpha_k = alpha_k_plus_1;
    npas = npas + 1;
end
```

Sur la distribution de données assez bien séparée que nous avons générée on peut voir que la méthode linéaire marche parfaitement. La marge est en effet maximale et les deux ensembles bien séparés.



La méthode devient tout de suite moins efficace voire complètement fausse lorsque les données ne sont pas distribuées de façon à avoir deux familles séparables par une droite avec une marge conséquente. On peut le voir sur les deux exemples suivants :





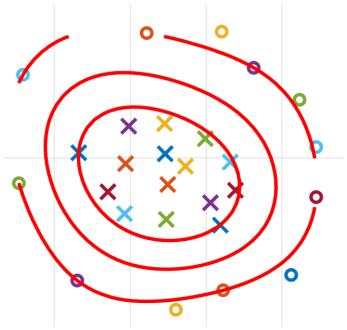
Dans ce cas, on considère une version plus généralisée de l'algorithme en utilisant les espaces de Hilbert.

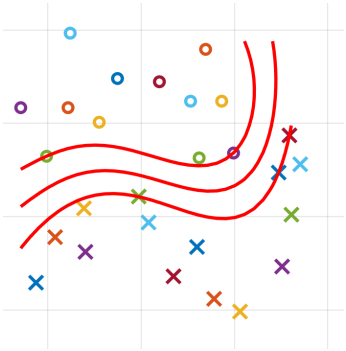
3 Cas non linéaire:

Dans le cas où les deux populations ne sont pas séparables par un hyperplan, l'idée est de plonger les x_k dans un espace de dimension plus grande où là, ils pourront être séparables par un hyperplan. En pratique, cet espace sera un espace de Hilbert dans lequel le produit scalaire est défini à partir d'un noyau.

L'implémentation de ce changement de dimension est très facile puisqu'il suffit de définir un nouveau produit scalaire gaussien de la forme : $K(x,y) = \exp(-\frac{\|x-y\|^2}{\sigma^2})$

La fonction kernel est déjà implémentée donc il suffit de changer la valeur de ikernel de 1 à 2 dans notre Main. On peut voir que la classification est beaucoup plus satisfaisante par rapport à la méthode linéaire.

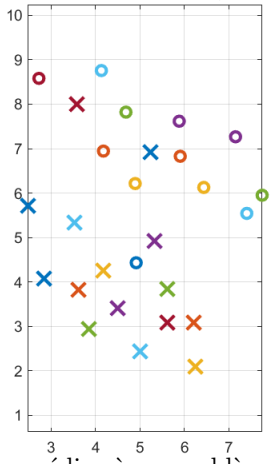


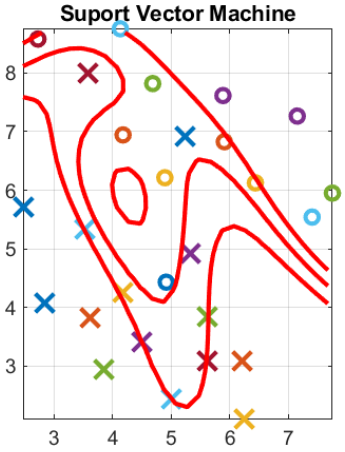


4 Introduction des marges souples :

En présence de données mal étiquetées (outliers), la technique des SVM vue dans les paragraphes précédents ne permet pas de trouver l'hyperplan optimal pour classifier les données.

Pour illustrer ceci, prenons l'ensemble des données 'data4.mat' sur lesquelles on applique l'algorithme codé précédemment. On voit dans la figure ci-dessous qu'il existe trois outliers, et il est clair que l'hyperplan séparateur obtenu n'est pas optimal. Dans notre cas c'est grave car une cellule cancéreuse peut-être considérée comme saine, chose qui peut avoir des conséquences fatales pour le moins...





Donc pour remédier à ce problème d'outliers, il est préférable d'utiliser la technique des SVM à marge souple qui tolère les observations mal classées. L'idée des SVM à marge souple est d'introduire pour chaque point une variable d'écart positive $\{\xi_k\}_{k=1..p}$ qui joue le rôle d'erreur et puis on pénalise ces erreurs dans le problème d'optimisation afin de trouver l'hyperplan séparateur optimal qui prend en compte les outliers. Donc, on est devant un nouveau problème d'optimisation sous contraintes dont la formulation primale est la suivante :

$$\min(\tfrac{1}{2}\|w\|^2 + C\sum_{k=0}^{p}\xi_k)$$
$$l_k(< w, \hat{x_k} > +b) \geq 1 - \xi_k \text{ avec } 1 \leq k \leq p$$

La condition dual change et devient :

$$\begin{cases} Max_{\alpha \geq 0}(H(\alpha)) \\ \sum_{k=0}^{p} \alpha_k l_k = 0 \\ 0 \leq \alpha \leq C \end{cases}$$

$$\text{avec } H(\alpha) = \frac{-1}{2} {}^T\alpha A\alpha + {}^T u\alpha$$

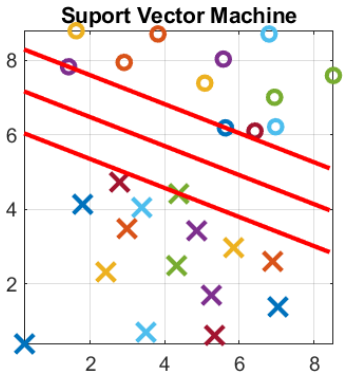
Alors, d'après la formulation duale, la seul modification à ajouter est la projection du α sur la boite $[0, C]^p$ au lieu du quadrant positif. Ainsi, pour le calcule du b, on ne prend que les points dont les $\alpha_i \neq C$ car pour ces points on a les $\xi_i = 0$ et donc il sera possible de calculer b avec l'equation suivante :

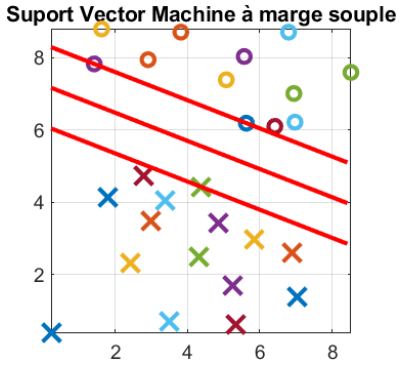
$1 - \xi_i - l_i(< w, \hat{x_i} > +b) = 0.$

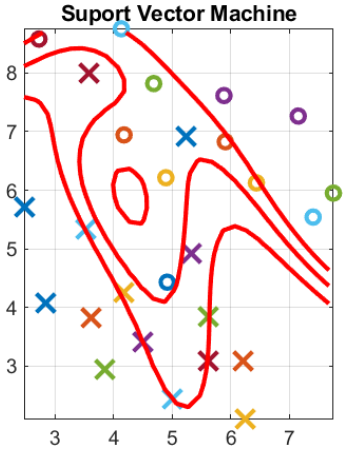
Pour le code matlab, il suffit d'ajouter les lignes suivantes à la boucle du gradient projeté :

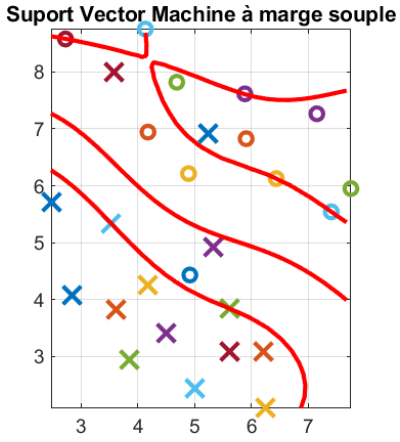
```
%critère de marge souple apres cela on a apla entre 0 et C
if marge_souple== true
    alph=min(alph,C_souple);
end
```

Maintenant, on applique le nouveau code sur les deux ensembles de données "data1.mat" et "data4.mat". Pour "data1.mat" (Les deux figures suivantes), on voit que l'hyperplan obtenu par les SVM et celui obtenu les SVM à marge souple sont identiques car tous les points de données sont bien placés et donc les variables d'écart positif sont toutes nuls. En revanche, pour "data4.mat", on voit que les deux techniques donnent des hyperplans différents et cela est dû à l'existence des outliers. Ainsi, l'hyperplan obtenu par les SVM à marges souples est bien meilleur et sépare bien les données.









5 Test de l'influence de quelques paramètres :

Cette section présente l'influence des paramètres SVM sur les performances de classification notamment le paramètre C et les paramètres de la fonction du noyau. Pour cette fin, différents tests ont été menés pour montrer l'influence de ces paramètres sur les performances de classification des SVM. Dans ces expériences, un simple exemple de classification binaire bidimensionnelle est proposé pour visualiser les échantillons correctement classés, les échantillons mal classés, la limite de décision et les bordures de marge. Nous avons utilisé deux échantillons de données dans ces tests. La première data (data1) était séparable linéairement et elle a été utilisée dans le test du noyau linéaire. Les autres données n'étaient pas séparables linéairement, et elles ont été utilisées pour différents noyaux, c'est-à-dire des fonctions de noyau non linéaires. Nous avons utilisé dans ce cas différents ensembles de données.

5.1 Fonction noyau linéaire :

Les classifieurs SVM avec noyau linéaire sont les plus simple à utiliser car on est dans le cas d'un produit scalaire normal. Ces classifieurs ne prennent pas de paramètres si ce n'est le paramètre C dans le cas d'un classifieur SVM à marges souples. Influence de C :

Illustration avec des données linéairement séparables :

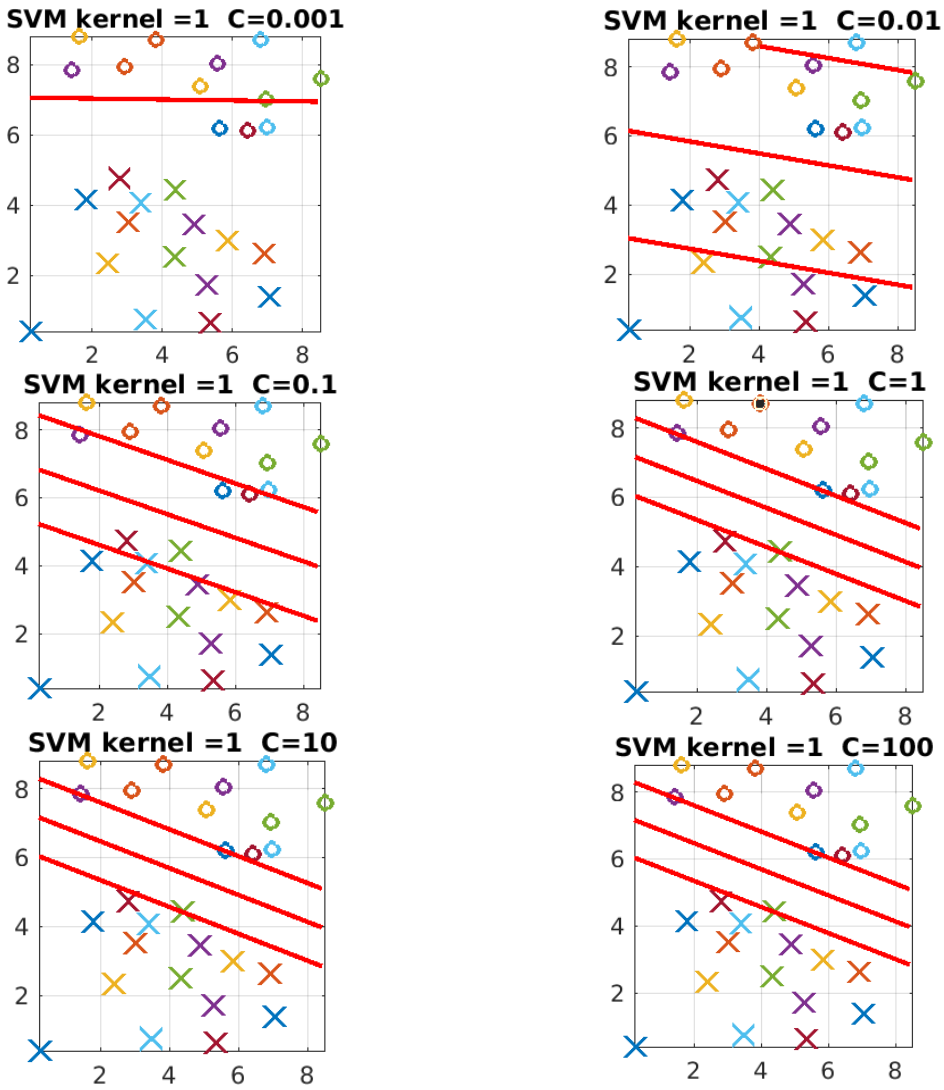


Figure 1 : Influence du paramètre C sur la performance des SVM linéaires.

Les données sont linéairement séparables pour des valeurs de C dans {0.001, 0.01, 0.1, 1, 10, 100}

Une petite valeur de C augmente le nombre de points mal classés et ce nombre diminue lorsque C augmente. La valeur de C est donc inversement proportionnelle à la largeur de la marge SVM : la largeur de marge maximale a été obtenue lorsque la valeur de C était minimale, et vice versa.

De petites valeurs de C (dans notre cas 0.001, 0.01, 0.1) augmentent la marge de SVM et par conséquent augmentent le nombre de points support qui peuvent conduire à un sous-ajustement sévère.

Réciproquement, de grandes valeurs de C (dans notre exemple C = 100) minimisent la largeur de la marge de SVM et augmentent le poids des échantillons non séparables. Une valeur aberrante, un bruit ou un échantillon critique peuvent alors déterminer la limite de décision, ce qui rend le classificateur sensible au bruit dans les données. Par conséquent, une valeur élevée de C peut entraîner un surajustement sévère.

Pour conclure, en fonction des données, la valeur de C peut varier dans un large intervalle et la valeur optimale de C peut être obtenue en essayant un nombre fini de valeurs pour rechercher celle qui produit l'erreur de classification minimale.

5.2 Fonction de noyau non-linéaire :

Noyau dans la fonction prédéfinie dans le code : $kern(u, v) = exp(-0.1 * \|u - v\|^2)$

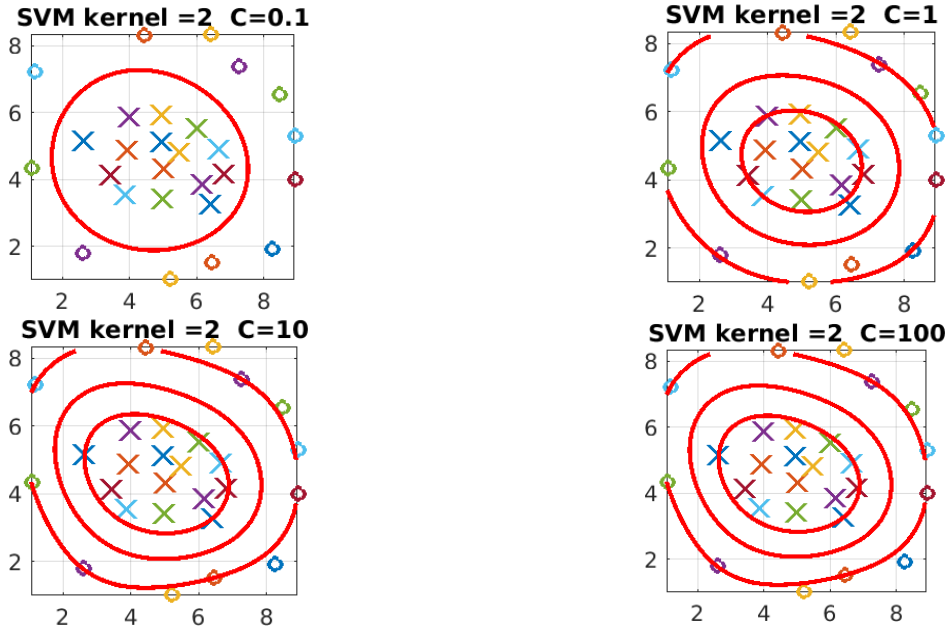


Figure 2 : influence du paramètre C sur la performance d'un SVM non linéaire

Avec une petite valeur du paramètre C (dans notre exemple C = 0,1 et C = 1), la largeur de la marge a été maximisée. Comme précédemment, de petites valeurs de C augmentent la marge de SVM et donc le nombre points support qui peuvent conduire à un sous-ajustement sévère. A contrario, pour de grandes valeurs de C (dans notre exemple C = 10, 100 ...) la largeur de la marge de SVM cesse de diminuer à partir d'un certain rang, donc on ne peut pas se tromper avec une grande valeur de C avec ce noyau. Toutefois, nous pouvons jeter un coup d'œil sur epsilon (seuil de convergence):

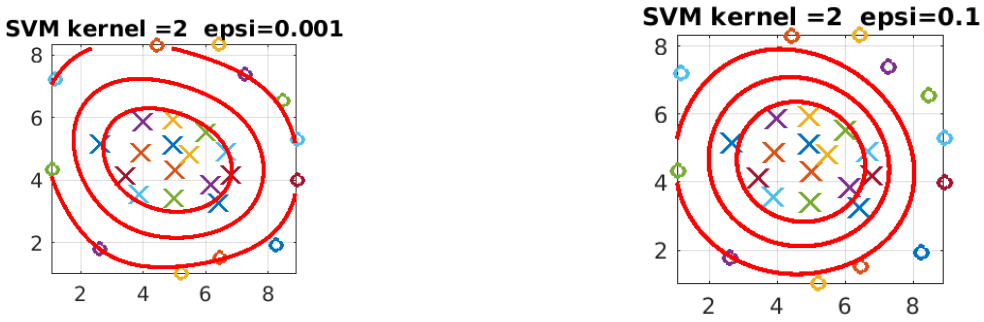


Figure 3 : L'influence du paramètre epsilon sur la performance d'un SVM non linéaire

Avec une petite valeur d'epsilon (dans notre exemple epsi = 1e-5), la largeur de la marge de SVM cesse d'augmenter à partir d'une certaine valeur, on ne peut donc pas nous tromper avec une petite valeur d'epsilon avec ce noyau.

En revanche, les grandes valeurs d'epsilon (dans notre exemple epsi = 0,1) minimisent la largeur de la marge de SVM et augmentent le poids des points non séparables, ce qui rend le classificateur sensible au bruit dans les données. Par conséquent, une valeur élevée d'epsilon peut entraîner un surajustement sévère.

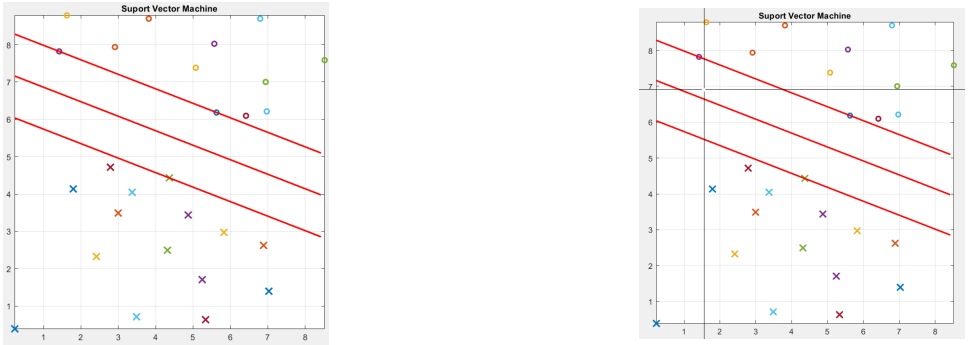
6 L'ajout d'un point pour tester le pouvoir classifieur de notre algorithme :

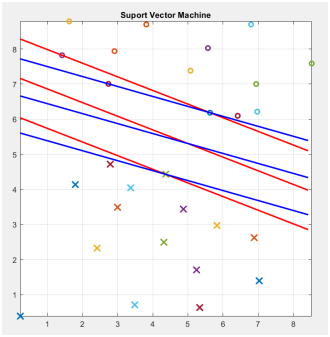
Quand on ajoute un nouveau point x_{new} dans le graph, l'idée pour lui attribuer un label s'inspire un peu de l'algorithme kNN.

En effet, on trace un cercle de rayon R - qu'on choisit judicieusement- et de centre le nouveau point x_{new} choisit. Ensuite on compte le nombre de points de label 1 présents dans le cercle, et on fait de même pour les points de label -1.

S'il y a plus de points de label 1, c'est qu'on est très probablement dans la zone de la classe de label 1 et on attribut par conséquent au nouveau point x_{new} le label 1. Et vice versa s'il y a dans le cercle plus de points de label – 1 présents.

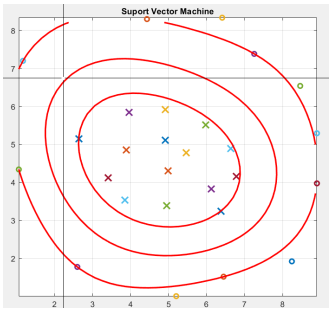
Le choix du rayon R dépend fortement de la densité des points dans le graphe. Par conséquent, avant de choisir le rayon R. On trace les données et on calcule la distance $(x_1 - x_{new1})^2 + (x_2 - x_{new2})^2$ pour des points x aléatoires histoire d'avoir une idée de l'ordre de grandeur du rayon à choisir.

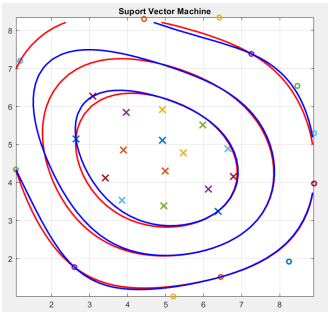


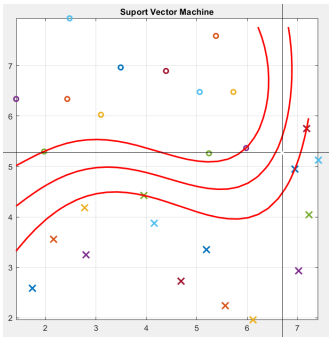


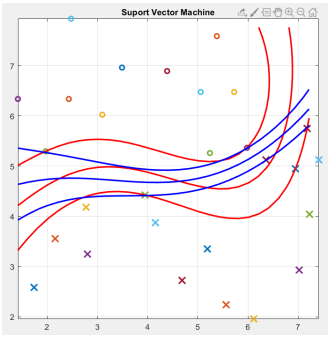
Pour répondre à la question de cette partie, le programme a la capacité d'attribuer un bon label en fonction de si le rayon est bien choisi.

Voici ci-dessus le retraçage de l'hyperplan et des marges pour différents types de classes :

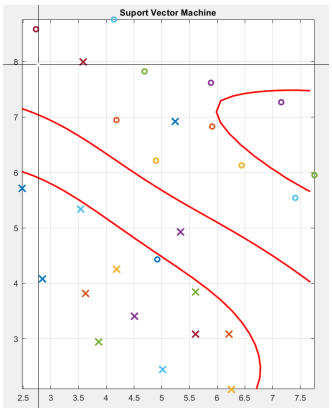


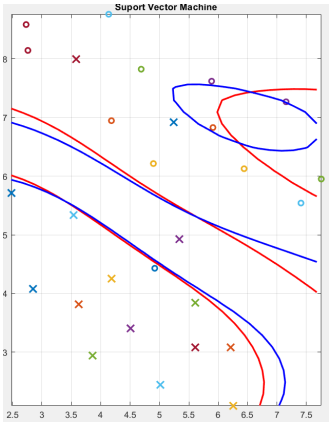






Ici on a choisi une zone critique ou les deux classes sont très proches l'une des l'autre et on voit bien que seul la densité détermine le label. Bien que le nouveau point choisi soit plus proche du point de label 1, la densité des points de label -1 et plus grande et par conséquent, le nouveau point x_{new} est labelisé -1.





7 Conclusion :

La machine à vecteurs de support (SVM) est l'un des algorithmes d'apprentissage des plus connus et a été utilisé dans de nombreuses applications dans divers domaines des sciences. Les SVM ont un paramètre C et des paramètres de noyau. La complexité et les performances du modèle de classification dépendent fortement du choix de ces paramètres, ce qui peut conduire à un modèle avec tendance de sous-ajustement ou surajustement sévère.