SÉPARATEURS À VASTE MARGE

Analyse de Données - TD 3

Université de Lorraine - ENSEM ISN 2A

FRANÇA DE SALES Déric Augusto

32219632

deric-augusto.franca-de-sales6@etu.univ-lorraine.fr

FERREIRA MARTINS Michelle

32219634

michelle.ferreira-martins5@etu.univ-lorraine.fr

07 janvier 2023

1 Introduction

Les machines à vecteurs de support ou séparateurs à vaste marge (en anglais *support-vector machine*, SVM) sont un ensemble de méthodes d'apprentissage supervisé qui analysent les données et reconnaissent les modèles, utilisés pour la classification et l'analyse de régression.

Ces algorithmes sont capables de diviser des points dans l'espace en deux catégories ou plus, à partir d'un ensemble de données de référence qui entraînent le modèle à appartenir à chaque catégorie. Dans cette pratique, nous trouverons des lignes de séparation entre les données qui les séparent en deux classes. Les algorithmes générés dans le MATLAB v.R2020a commentés sont joints.

Cette ligne cherchera à maximiser la distance entre les points les plus proches par rapport à chacune des classes. Cette distance entre l'hyperplan et le premier point de chaque classe est généralement appelée marge. Le SVM effectue d'abord la classification des classes, définissant ainsi chaque point appartenant à chacune des classes, puis maximise la marge. C'est-à-dire qu'il commence par classer correctement les classes puis, en fonction de cette restriction, définit la distance entre les marges.

2 Objectifs

Générer des données linéairement séparables et non séparables et tracer la ligne séparatrice, les vecteurs de support et les lignes de vecteur de support pour chacune des données.

3 Génération des données linéairement séparables

Tout d'abord, deux matrices de données sont générées dans R^2 : X1 et X2. Ces matrices de données initiales seront linéairement séparables, c'est-à-dire qu'elles ne présenteront pas de points communs dans leurs limites de domaine. Les données pour X1 et X2 sont définies respectivement par (1) et (2) et représentées par les figures 1 et 2. Chaque matrice créée présente 10 points aléatoires dans le domaine défini.

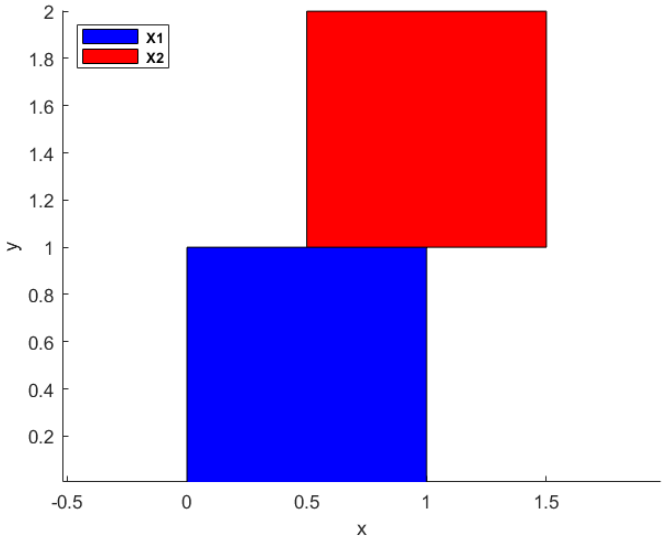

Figura 1: Espace limite des observations aléatoires générées X1 et X2.

$$X1 = (x, y) \in R^2; \;\; 0 < x < 1 \;\; \text{et} \;\; 0 < y < 1 \tag{1}$$

$$X2 = (x, y) \in R^2; \;\; 0.5 < x < 1.5 \;\; \text{et} \;\; 1 < y < 2 \tag{2}$$

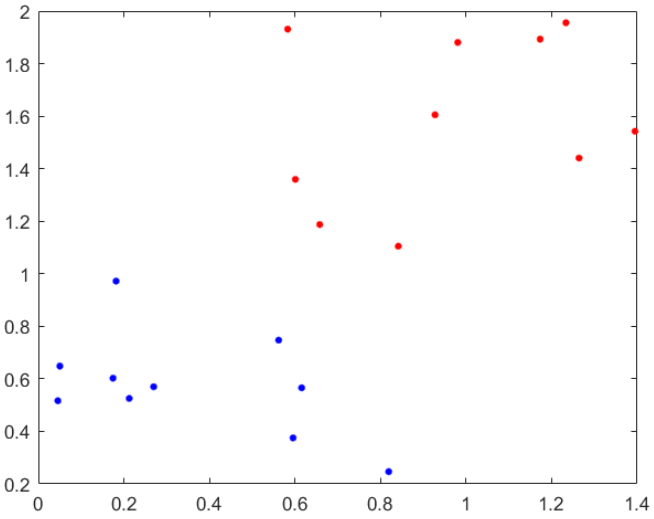

Figura 2: Observations aléatoires générées X1 et X2.

4 Élaboration du séparateur à vaste marge

Notre objectif ici est de séparer les données X1 et X2 avec une séparatrice dont la marge est maximisée. C'est un problème d'optimisation primal. Cette séparatrice à vaste marge linéaire (SVM) est un discriminante linéaire sous la forme : $D(x) = signe(wTx + b)$ où $w \in R_p$ et $b \in R$.

w et b sont donnés par la résolution du problème d'optimisation sous contraintes suivant :

$$\text{Primal} \quad \begin{cases} \min\limits_{w,b} \dfrac{1}{2}||w||^2 \\ \text{sous } \ell_k(w^T x_k + b) \geq 1, \quad k = 1, \ldots, N \end{cases}$$

Le problème d'optimisation sous contraintes précédent est un "programme quadratique" de la forme générale :

$$\begin{cases} \min\limits_{z} \dfrac{1}{2} z^T H z + f^T z \\ \text{sous } Az \leq e \end{cases}$$

avec

$$z = \begin{pmatrix} w \\ b \end{pmatrix} \in \mathbb{R}^{p+1} \qquad f = (0, \ldots, 0)^T \in \mathbb{R}^{p+1} \qquad H = \begin{pmatrix} I_p & 0 \\ 0 & 0 \end{pmatrix}$$

$$A = -(\text{diag}(\ell)X \quad \ell) \qquad e = -(1, \ldots, 1)^T \in \mathbb{R}^N \qquad \ell \in \mathbb{R}^N \text{ vecteur des signes}$$

$X \in \mathbb{R}^{N \times p}$ matrice dont la ligne k est x_k^T

où N est le nombre total de points et p, le nombre de rangs de données. Les autres matrices sont créées à partir de ces données, selon la modélisation ci-dessus et le problème pourrait

être résolu dans MATLAB au moyen de l'outil *quadprog*, qui reçoit en entrée les matrices et le vecteur H, f A et e et renvoie le vecteur w et la constante b. En utilisant les valeurs obtenues (selon la relation (3)), il est alors possible de tracer la séparatrice, comme le fait la figure 3.

$$w^T \cdot x + b = 0 \implies y_{\text{séparatrice}} = -\frac{w_1}{w_2} \cdot x - \frac{b}{w_2}; \quad w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \tag{3}$$

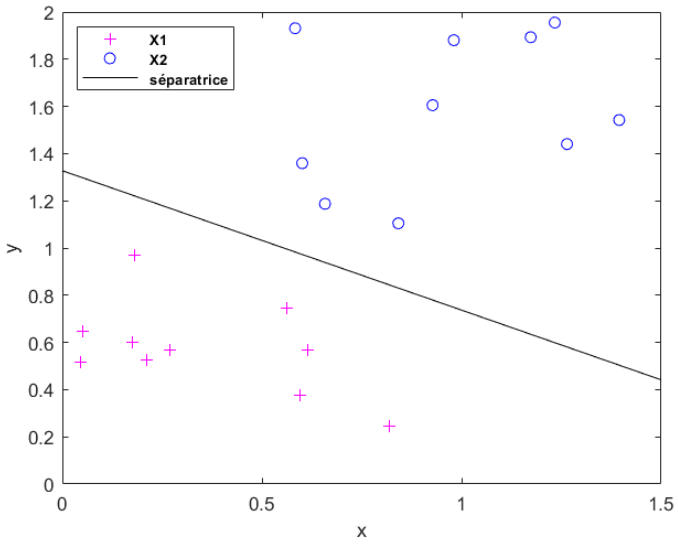

Figura 3: Donnés X1 et X2 séparés par la ligne séparatrice.

Un autre retour de la fonction *quadriprog* est la variable lambda, qui contient les multiplicateurs de Lagrange trouvés dans l'optimisation du problème. Grâce à cette variable, il est alors possible de créer la matrice $Xmarq$ contenant les vecteurs de support.

Ces vecteurs supports, à leur tour, représentent les points où les paramètres de Lagrange sont nuls (approximativement zéro dans le calcul numérique). Ainsi, ils ont pu être tracés et représentés dans la figure 4.

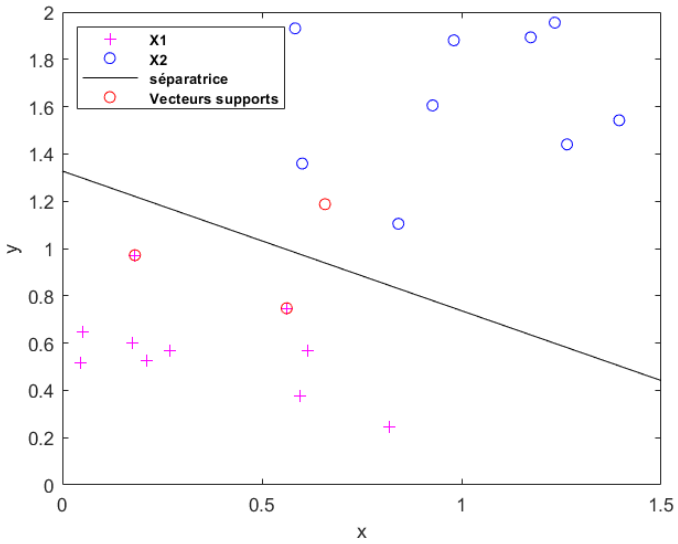

Figura 4: Donnés X1 et X2 séparés par la ligne séparatrice, mettant en évidence les vecteurs de support.

De plus, à travers les vecteurs supports trouvés, il est possible de tracer les lignes de marge géométriques, qui sont parallèles à la séparatrice et passent par les vecteurs supports. Sachant que la distance entre les points trouvés des vecteurs supports et la séparatrice sera de $1/|w|$, il a été possible de dessiner les marges géométriques de la figure 5.

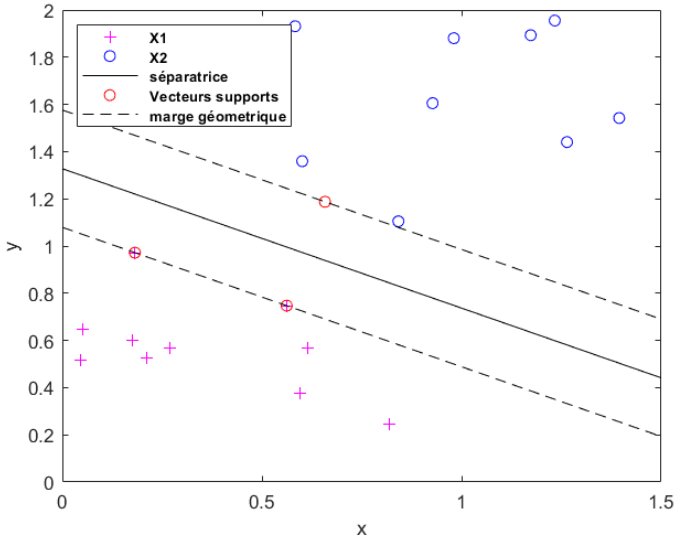

Figura 5: Donnés X1 et X2 séparés par la ligne séparatrice, avec la marge géométrique des vecteurs supports.

Nous avons ensuite pu dessiner la séparatrice, ainsi que les droites et les arêtes géométriques qui lui sont parallèles et passent par les vecteurs supports. Vous pouvez voir que les vecteurs de support sont les données dont la distance par rapport à la séparatrice est la minimale pour chaque type de données.

5 Génération des données non linéairement séparables

Comme précédemment, nous allons générer les matrices de données X3 et X4, à la différence près que leurs domaines se croisent dans la région contenue dans (4). X3 et X4 sont définis comme (5) et (6) respectivement et représentés par les figures 6 et 7.

$$(x, y) \in R^2; \;\; 0.8 < x < 1 \;\; \text{et} \;\; 0.8 < y < 1 \tag{4}$$

$$X3 = (x, y) \in R^2; \;\; 0 < x < 1 \;\; \text{et} \;\; 0 < y < 1 \tag{5}$$

$$X4 = (x, y) \in R^2; \;\; 0.8 < x < 1.8 \;\; \text{et} \;\; 0.8 < y < 1.8 \tag{6}$$

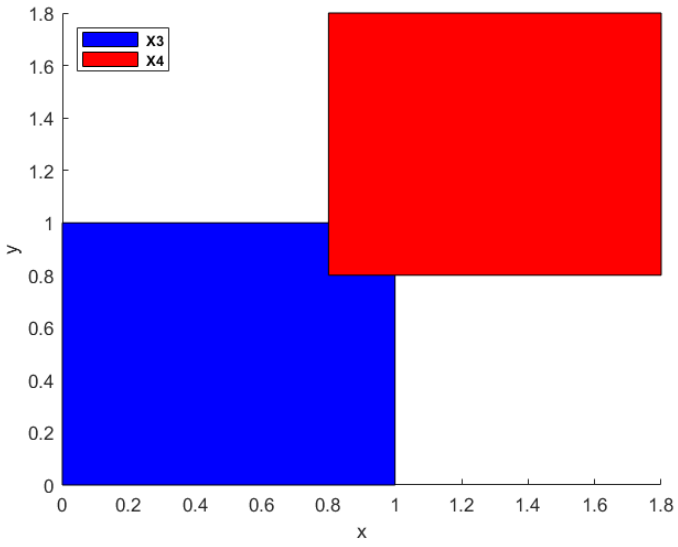

Figura 6: Espace limite des observations aléatoires générées X3 et X4.

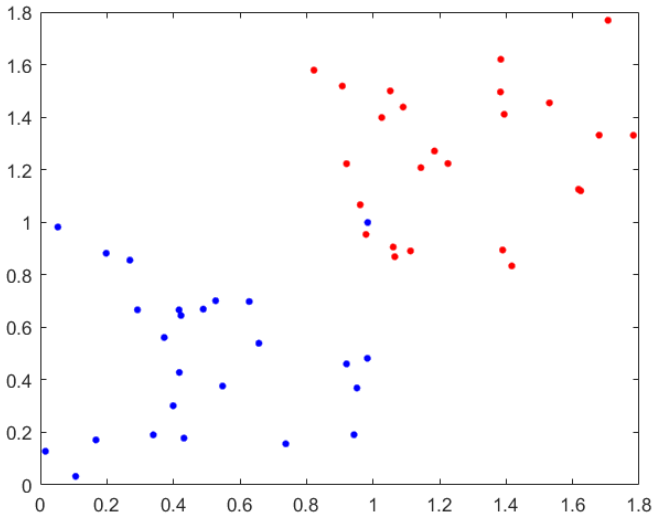

Figura 7: Observations aléatoires générées X3 et X4.

6 Élaboration du séparateur à marge poreuse

Maintenant, nous avons un défi supplémentaire pour dessiner la séparatrice puisque les domaines interviennent. Le problème d'optimisation évolue alors sous la forme suivante :

$$\begin{cases} \min\limits_{w,b,\xi_k} \dfrac{1}{2}\|w\|^2 + C\sum\limits_{k=1}^{N}\xi_k \\ \text{sous} \quad \ell_k(w^T x_k + b) \geq 1 - \xi_k, \quad k = 1, \ldots, N \\ \qquad\quad \xi_k \geq 0, \quad k = 1, \ldots, N \end{cases}$$

où le scalaire C permet de quantifier les poids respectifs de la marge que l'on cherche à maximiser, et ξ_k la somme des variables des ressorts, que on cherche à minimiser. Ainsi, comme réalisé précédemment, on obtient les figures 8, 9 et 10. Cette fois, nous considérerons tous les points comme des vecteurs de support, en raison des nouvelles caractéristiques du problème.

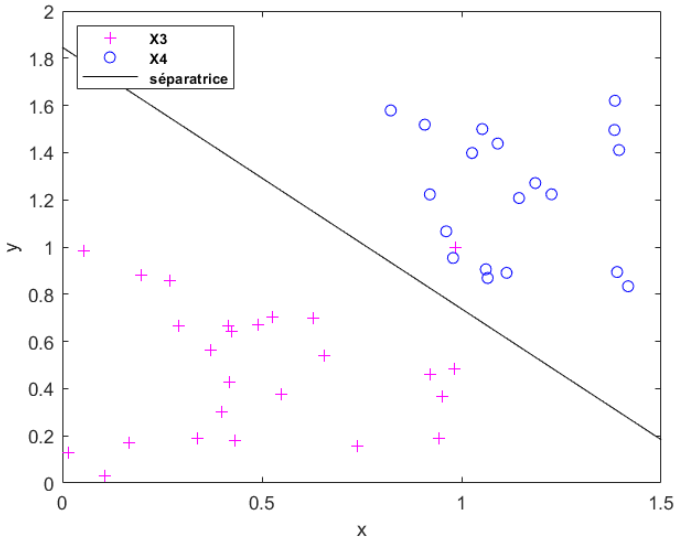

Figura 8: Donnés X3 et X4 séparés par la ligne séparatrice.

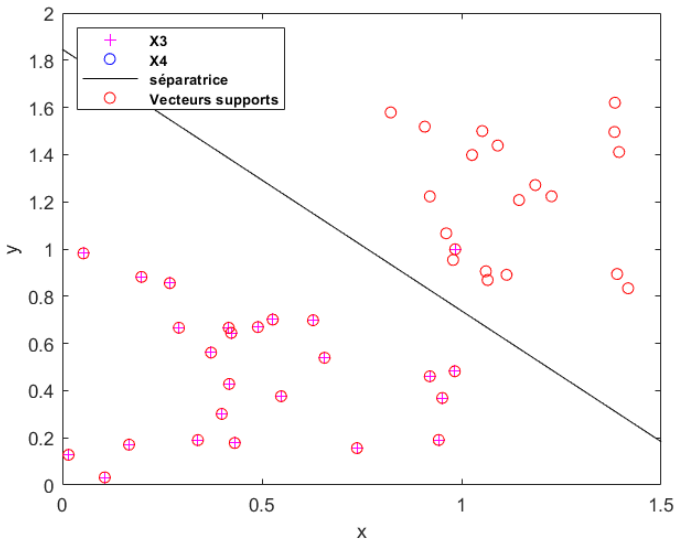

Figura 9: Donnés X3 et X4 séparés par la ligne séparatrice, mettant en évidence les vecteurs de support.

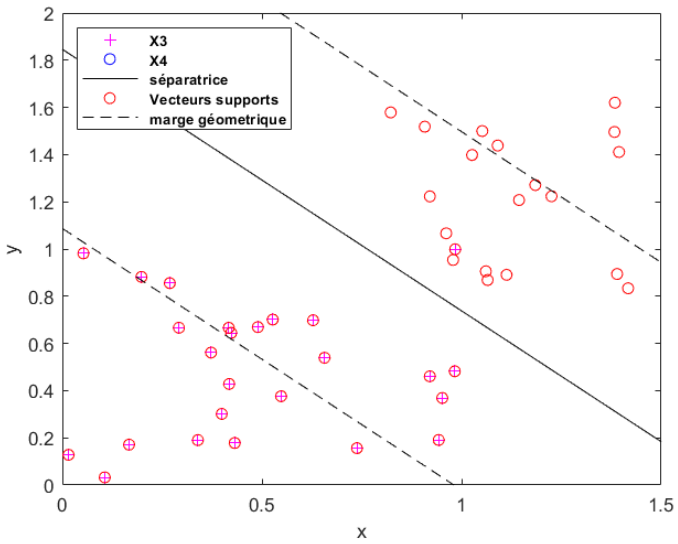

Figura 10: Donnés X3 et X4 séparés par la ligne séparatrice, avec la marge géométrique des vecteurs supports.

Dans le type de données visé, la séparatrice ne sera pas toujours correcte, car un dé peut s'écarter de la région séparée. De même, nous observons que la marge géométrique ne nous donne pas la même signification par rapport au problème précédent.

Annexe

12/12/2022

realisé en MATLAB v.R2020a.

binôme :

- Déric Augusto França de Sales
- Michelle Ferreira Martins

sommaire


```
close all; clear all; clc;
```

1 Génération des données linéairement séparables

1.1 Espace limite des observations aléatoires générées X1 e X2

```
% Génération de l'image des carrés
squareX1 = [0, 1, 1, 0; 0, 0, 1, 1];
squareX2 = [0.5, 1.5, 1.5, 0.5; 1, 1, 2, 2];

patch(squareX1(1, :), squareX1(2, :), 'b')
patch(squareX2(1, :), squareX2(2, :),'r');
xlim([-0.5, 2]);
ylim([0, 2]);
legend({' X1',' X2'}, 'Location', 'northwest', 'FontSize', 8, 'FontWeight', 'bold')
xlabel('x')
ylabel('y')
```

1.2 Tracer les donnés X1 e X2

```
% Matrice X1 avec des valeurs variant de 0 à 1 pour les abscisses et les ordonnées
X1 = rand(10, 2);
figure
plot(X1(:, 1), X1(:, 2), 'b.', 'Markersize', 12);
hold on
% Matrice X2 avec des valeurs variant de 0,5 à 1,5 pour les abscisses et de 1 à 2
%pour les ordonnées
X2 = [0.5 + rand(10,1), 1 + rand(10,1)];
plot(X2(:, 1), X2(:, 2), 'r.', 'Markersize', 12);
X = [X1; X2]; % Matrice 20x2
l = [ones(1,10), -ones(1, 10)]; % Vetor l
```

2 Elaboration du separateur a vaste marge

2.1 Construtions des matrices d'optimisation.

Utiliser la fontion *quadprog* pour resoudre le problème d'optimisation.

```
% Matrices

N = 20;
p = 2;
H = eye(p);
H(:, end + 1) = 0;
H(end + 1, :) = 0;
f = zeros(1, p + 1)';
A = -[diag(l)*X, l'];
e = -ones(1, N)';

% Optimisation

[z, fval, exitflag, output, lambda] = quadprog(H,f,A,e);
% La fonction quadprog renvoie une matrice 3x1 z où les deux premières lignes sont w
%et la troisième est la constante b
w = z(1:2);
b = z(3);
```

2.2 Tracer la separatrice

```
a = (w(1)/(-w(2)));
b = b/(-w(2));
y0 = a*0 + b;
y1_5 = a*1.5 + b;
droiteX = [0 1.5];
droiteY = [y0 y1_5];

figure
plot(X(1:10,1), X(1:10,2), '+m', X(11:20,1), X(11:20,2), 'ob', droiteX, droiteY, 'k');
xlim([0, 1.5]);
ylim([0,2]);
```

```
legend({'  X1', '  X2', '  séparatrice'}, 'Location', 'northwest', 'FontSize', 8, ...
'FontWeight', 'bold')
xlabel('x')
ylabel('y')
```

2.3 Détecter les observations supports et superposer, sur la représentation graphique un marqueur supplémentaire

```
support_vector = find(lambda.ineqlin > 1e-3);
supports_points = X(support_vector(:), :);
figure
plot(X(1:10,1), X(1:10,2), '+m',  X(11:20,1), X(11:20,2), 'ob', droiteX, droiteY, ...
    'k', supports_points(:,1), supports_points(:,2), 'or');
xlim([0, 1.5]);
ylim([0,2]);
legend({'  X1', '  X2', '  séparatrice', '  Vecteurs supports'}, 'Location', ...
    'northwest', 'FontSize', 8, 'FontWeight', 'bold')
xlabel('x')
ylabel('y')
```

2.4 Déterminer l'équation des droites parallèles à la séparatrice matérialisant la marge et les tracer en pointillés.

```
distance = 1/abs(w(2));
droiteYsup = droiteY + 1/abs(w(2));
droiteYinf = droiteY - 1/abs(w(2));
figure
plot(X(1:10,1), X(1:10,2), '+m',  X(11:20,1), X(11:20,2), 'ob', droiteX, droiteY, ...
    'k', supports_points(:,1), supports_points(:,2), 'or', droiteX, droiteYsup, ...
    '--k', droiteX, droiteYinf, '--k');
xlim([0, 1.5]);
ylim([0,2]);
legend({'  X1', '  X2', '  séparatrice', '  Vecteurs supports', ...
    '  marge géometrique'}, 'Location', 'northwest', 'FontSize', 8, ...
    'FontWeight', 'bold')
xlabel('x')
ylabel('y')
```

3 Génération des données non linéairement séparables

3.1 Espace limite des observations aléatoires générées X3 e X4

```
% Génération de l'image des carrés
squareX3 = [0, 1, 1, 0; 0, 0, 1, 1];
squareX4 = [0.8, 1.8, 1.8, 0.8; 0.8, 0.8, 1.8, 1.8];
figure
patch(squareX3(1, :), squareX3(2, :), 'b')
patch(squareX4(1, :), squareX4(2, :),'r');
xlim([0, 1.8]);
ylim([0, 1.8]);
legend({' X3',' X4'}, 'Location', 'northwest', 'FontSize', 8, 'FontWeight', 'bold')
xlabel('x')
```

```
ylabel('y')
```

3.2 Tracer les donnés X3 e X4

```
% Matrice X1 avec des valeurs variant de 0 à 1 pour les abscisses et les ordonnées
X3 = rand(25, 2);
figure
plot(X3(:, 1), X3(:, 2), 'b.', 'Markersize', 12);
hold on
% Matrice X2 avec des valeurs variant de 0,5 à 1,5 pour les abscisses et
%de 1 à 2 pour les ordonnées
X4 = [0.8 + rand(25,1), 0.8 + rand(25,1)];
plot(X4(:, 1), X4(:, 2), 'r.', 'Markersize', 12);
```

4 Elaboration du séparateur à marge poreuse

4.1 Construtions des matrices d'optimisation.

```
X = [X3; X4]; % Matrice 50x2
l = [ones(1,25), -ones(1, 25)]; % Vetor l

% Matrices

N = 50;
p = 2;
H = eye(p);
H(:, end + 1) = 0;
H(end + 1, :) = 0;
f = zeros(1, p + 1)';
A = -[diag(l)*X, l'];
e = -ones(1, N)';

% Optimisation

[z, fval, exitflag, output, lambda] = quadprog(H,f,A,e);
% La fonction quadprog renvoie une matrice 3x1 z où les deux premières lignes
%sont w et la troisième est la constante b
w = z(1:2);
b = z(3);
```

4.2 Reprendre les questions 2 à 4 du cas des données séparables linéairement.

```
% Tracer la séparatrice

a = (w(1)/(-w(2)));
b = b/(-w(2));
y0 = a*0 + b;
y1_5 = a*1.5 + b;
droiteX = [0 1.5];
droiteY = [y0 y1_5];

figure
```

```
plot(X(1:25,1), X(1:25,2), '+m', X(26:50,1), X(26:50,2), 'ob', droiteX, droiteY, 'k');
xlim([0, 1.5]);
ylim([0,2]);
legend({'  X3', '  X4', '  séparatrice'}, 'Location', 'northwest', 'FontSize',...
    8, 'FontWeight', 'bold')
xlabel('x')
ylabel('y')
```

4.3 Détecter les observations supports et superposer, sur la représentation graphique un marqueur supplémentaire

```
support_vector = find(lambda.ineqlin > 1e-3);
supports_points = X(support_vector(:), :);
figure
plot(X(1:25,1), X(1:25,2), '+m',  X(26:50,1), X(26:50,2), 'ob', droiteX, ...
    droiteY, 'k', supports_points(:,1), supports_points(:,2), 'or');
xlim([0, 1.5]);
ylim([0,2]);
legend({'  X3', '  X4', '  séparatrice', '  Vecteurs supports'}, 'Location', ...
    'northwest', 'FontSize', 8, 'FontWeight', 'bold')
xlabel('x')
ylabel('y')
```

4.4 Déterminer l.équation des droites parallèles à la séparatrice matérialisant la marge et les tracer en pointillés

```
distance = 1/abs(w(2));
droiteYsup = droiteY + 1/abs(w(2));
droiteYinf = droiteY - 1/abs(w(2));
figure
plot(X(1:25,1), X(1:25,2), '+m',  X(26:50,1), X(26:50,2), 'ob', droiteX, droiteY,...
    'k', supports_points(:,1), supports_points(:,2), 'or', droiteX, droiteYsup, ...
    '--k', droiteX, droiteYinf, '--k');
xlim([0, 1.5]);
ylim([0,2]);
legend({'  X3', '  X4', '  séparatrice', '  Vecteurs supports', ...
    '  marge géometrique'}, 'Location', 'northwest', 'FontSize', 8, ...
    'FontWeight', 'bold')
xlabel('x')
ylabel('y')
```