

# Rapport sur le projet d'Optimisation Support-Vector Machines

Kawisorn Kamtue & Clémence Réda

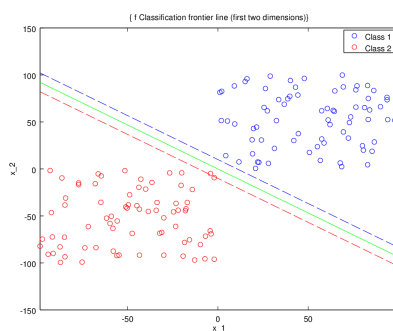
December 30, 2016

## 1 Support Vector Machine

Les *Support Vector Machine solvers* (SVM) sont une catégorie d'algorithmes d'apprentissage statistique supervisé. Ils permettent de résoudre le problème de classification binaire suivant :

Etant donnés  $(x_i)_{i \leq m}$  des points dans  $\mathbb{R}^n$ , et  $(y_i)_{i \leq m}$  les étiquettes des points tels que l'étiquette de  $x_i$  soit  $y_i \in \{-1, 1\}$ , on cherche la droite qui sépare "le mieux possible" les points dans différentes classes, autrement dit, la frontière de Voronoi entre les deux classes.

Figure 1: Exemple de frontière pour deux classes : celles des points bleus et celle des points rouges



La frontière que l'on recherche est une fonction linéaire, donc de la forme (avec deux paramètres de dimension 1  $\omega$  et  $b$ ) :

$$f : X \rightarrow \omega^T X + b = \begin{bmatrix} \omega & b \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix}$$

telle que :

$$\begin{aligned}\forall i, y_i = -1 &\Rightarrow f(x_i) \leq -1 \\ \forall i, y_i = 1 &\Rightarrow f(x_i) \geq 1 \\ \Leftrightarrow \forall i, y_i \times f(x_i) &\geq 1 \quad (1)\end{aligned}$$

Pour simplifier le problème, on peut prendre  $\omega' = \begin{bmatrix} \omega \\ b \end{bmatrix}$  et  $x' = \begin{bmatrix} x \\ 1 \end{bmatrix}$  (que l'on notera par souci de simplicité  $\omega$  et  $x$ ).

Pour obtenir un résultat robuste, on souhaite que les deux droites  $f(X) = 1$  et  $f(X) = -1$  soient les plus distantes possibles. En effet, si ces deux droites sont trop proches, cela signifie que la probabilité d'erreur quant à la prédiction de la classe d'un point proche de ces droites sera importante.

La distance  $\gamma$  entre ces deux droites se calcule de la façon suivante : soient  $u, v$  deux points tels que  $f(v) = 1$  et  $f(u) = -1$ . Alors :

$$\|f(v) - f(u)\| = \|\omega \times (v - u)\| = \|\omega\| \times \|(v - u)\| = \|\omega\| \times \|\gamma\| = \|1 - (-1)\| = 2$$

Finalement, le problème d'optimisation à résoudre pourrait être :

$$\begin{aligned} \max_w \gamma &= \frac{2}{\|\omega\|} \text{ avec (1)} \\ &\Leftrightarrow \min_w \|\omega\| \text{ avec (1)} \\ &\Leftrightarrow \min_w \frac{1}{2} \times \|\omega\|^2 \text{ avec (1) pour faciliter les calculs} \end{aligned}$$

Un autre problème se pose si on s'arrête ici : par exemple, dans l'exemple de la frontière de Voronoi que l'on a vu ci-dessus, si on a un point bleu au milieu des points rouges, alors il n'existe pas de droite telle qu'il n'y ait que de points bleus d'un côté et que des points rouges de l'autre côté, ce qui contredit la condition (1). Le problème est alors infaisable. Pourtant, la droite dessinée en vert peut sembler acceptable comme frontière pour l'ensemble de points auquel on a ajouté un point bleu au milieu du nuage de points rouge.

On tient compte de cette erreur en introduisant les variables  $(z_i)_{i \leq m}$ . Pour que la condition (1) soit toujours vérifiée, il faut que quand  $y_i \times f(x_i) \geq 1$ ,  $z_i = 0$  et lorsque  $y_i \times f(x_i) < 1$ ,  $z_i = 1 - y_i \times f(x_i)$ . Le but étant de minimiser le nombre de ces erreurs, ie. points mal classés, on utilise un paramètre  $C$  constant qui permet d'insister plus ou moins sur la minimisation de ces erreurs :

$$\begin{aligned} \text{(P)} \quad \min_w \quad & \frac{1}{2} \times \|\omega\|^2 + C \times \sum_{i \leq m} z_i \\ \text{avec} \quad & \forall i, z_i \geq 0 \\ & \forall i, y_i \times (\omega^T x_i) \geq 1 - z_i \end{aligned}$$

Les fonctions que l'on a introduites sont toutes convexes. Si la dimension des points  $(x_i)_i$  est petite, nous allons pouvoir utiliser la méthode de Newton pour résoudre ce problème. On verra par la suite le *kernel trick* qui permettra de ne pas tenir compte de la dimension, mais seulement du nombre d'échantillons  $(x_i)_i$ .

## 2 Calcul du dual

Calculons le lagrangien du problème (P). Soit  $\lambda$  le multiplicateur de Lagrange de dimension  $1 \times m$ :

$$\begin{aligned} \forall w, \lambda \in \mathbb{R}^{2m}, L(\omega, \lambda, z) &= \\ &= \frac{1}{2} \|w\|^2 + C \times \sum_i z_i - \sum_i \lambda_i \times z_i + \sum_i \lambda_i \times (1 - y_i \omega^T x_i) \\ &= \frac{1}{2} \|w\|^2 + C \mathbf{1}^T z - C \lambda^T z + \mathbf{1}^T \lambda - \sum_i \lambda_i \times y_i \omega^T x_i \\ &= \frac{1}{2} (\|w - \sum_i \lambda_i y_i x_i\|_2^2 - \|\sum_i \lambda_i y_i x_i\|_2^2) + (C \mathbf{1} - \lambda)^T z + \mathbf{1}^T \lambda \end{aligned}$$

Minimisons L par rapport à  $\omega$ . Comme le lagrangien est convexe en  $\omega$ , il faut annuler le gradient :

$$\begin{aligned} \nabla_\omega L(\omega, \lambda, z) &= \frac{1}{2} (2\omega - 2 \sum_i \lambda_i y_i x_i) + 0 = 0 \\ \Leftrightarrow \omega &= \sum_i \lambda_i y_i x_i \quad (1) \end{aligned}$$

Minimisons L par rapport à  $z$ . Comme le lagrangien est convexe en  $z$ , il faut annuler le gradient :

$$\begin{aligned} \nabla_z L(\omega, \lambda, z) &= 0 + C \mathbf{1}_{z>0}^T - \lambda \mathbf{1}_{z>0}^T = 0 \\ \Leftrightarrow mC - \sum_i \lambda_i &= 0 \text{ si } z_i > 0 \\ \Leftrightarrow mC &= \sum_i \lambda_i \text{ si } z_i > 0 \end{aligned}$$

Le minimum en L par rapport à  $z$  a une valeur finie ssi  $C \mathbf{1} - \lambda = 0$ . On obtient le problème dual en injectant les valeurs de  $\omega$  et de  $z$  dans le lagrangien :

$$\begin{aligned} \max_{\lambda \in \mathbb{R}^{+m}} & - \frac{1}{2} \|\sum_i \lambda_i y_i x_i\|_2^2 + \mathbf{1}^T \lambda \text{ par (1)} \\ \text{avec } \forall i, & 0 \leq \lambda_i \leq C \text{ si } z_i > 0 \\ \text{(vient des conditions de KKT -} & \text{complementary slackness,} \\ & \text{vérifiées car le problème est convexe)} \end{aligned}$$

On obtient la solution optimale du primal  $(\omega^*, z^*)$  à partir de celle du dual  $\lambda^*$  :

$$(1) \quad \omega^* = \sum_i \lambda_i^* y_i x_i$$

## 3 Utilisation de l'astuce du noyau (*kernel trick*)

Pour pouvoir trouver efficacement la solution au problème avec la méthode de Newton, il faut s'affranchir de la contrainte quadratique sur la dimension des échantillons. On note  $X$  la matrice des échantillons, et la matrice du noyau  $K = X^T X$ , avec  $K \geq 0$ . On montre alors que le problème dual peut se réécrire de la façon suivante :

$$\begin{aligned} \max & -\frac{1}{2}\lambda^T \text{diag}(y)K\text{diag}(y)\lambda + \mathbf{1}^T \lambda \\ \text{avec } & \forall i, 0 \leq \lambda_i \leq C \end{aligned}$$

ce qui équivale à :

$$\begin{aligned} \min & \frac{1}{2}\lambda^T \text{diag}(y)K\text{diag}(y)\lambda - \mathbf{1}^T \lambda \\ \text{avec } & \forall i, 0 \leq \lambda_i \leq C \end{aligned}$$

On remarque que la dimension  $m$  des échantillons n'intervient plus, et que donc la complexité de la résolution du problème ne dépend que du nombre d'échantillons.

## 4 Méthode de la barrière logarithmique

Enfin, on peut s'affranchir des contraintes d'inégalité sur le multiplicateur de Lagrange  $\lambda$  en posant la fonction barrière suivante :

$$\Phi(\lambda) = \sum_i (-\log(C - \lambda_i) - \log(\lambda_i)) = \sum_i \log\left(\frac{1}{(C - \lambda_i)\lambda_i}\right) = -\sum_i \log((C - \lambda_i)\lambda_i)$$

Cette fonction vaut  $+\infty$  si  $a < 0$  ou  $a > C$ . Le problème à optimiser devient alors (en changeant le signe pour obtenir une fonction à minimiser) :

$$\begin{aligned} \min & \frac{1}{2}\lambda^T \text{diag}(y)K\text{diag}(y)\lambda - \mathbf{1}^T \lambda + \Phi(\lambda) \\ \text{avec } & \forall i, 0 \leq \lambda_i \leq C \end{aligned}$$

## 5 Résultats

### 5.1 Comparaison entre les différentes générations de points

#### 5.1.1 Tableau récapitulatif

$d$  est la dimension des points et  $n$  le nombre d'échantillons dans la génération. On utilise  $\frac{2}{3}$  des points (choisis au hasard uniformément) de l'ensemble de départ pour l'ensemble d'entraînement, et les points restants pour l'ensemble de validation.

Table 1: Comparaison entre les générations de points

DONNÉES	C	D	N	N IT.	TEMPS (s)	MEILLEUR C	ECHEC (%)
1	1	40	10	11	25,414	1	0 (*)
1	5	40	10	11	0,177	1	0 (*)
1	10	40	10	11	0,168	1	0 (*)
1	5	40000	10	11	0,315	X	0
1	5	40	100	12	0,715	X	0
1	5	40	1000	?	à 10	?	?
2	5	200	150	12	0,689	?	0
3	5	200	150	12	0,660	?	0
4	5	200	150	12	0,655	?	0
5	5	200	150	12	0,709	?	4

Quelques remarques :

- De manière générale, utiliser une valeur de  $C$  plus grande accélère considérablement la recherche de la solution duale : pas au niveau du nombre d'itérations de la méthode de Newton, mais au niveau du coût de l'appel à la méthode de Newton (voir les trois premiers tests).
- La complexité temporelle de la résolution du problème dual est bien indépendante de la dimension et dépendante de la taille de l'échantillon (voir les tests 4, 5 et 6).
- La valeur "?" signifie que l'algorithme a tourné trop longtemps pour la valeur soit mesurée.
- La notation "(\*)" signifie que les tests marqués ont utilisé le même ensemble de données.

### 5.1.2 Validation croisée pour le choix de la meilleure valeur de $C$

Les deux fonctions *choiceC* et *crossvalidation* permettent de sélectionner la meilleure valeur de  $C$  pour un échantillon donné, par la méthode de *leave-one-out*, où, pour un échantillon de taille  $n$ , à chaque itération on choisit un élément  $e$  comme ensemble de test, et l'entraînement du SVM se fait sur les  $n - 1$  éléments restants. La valeur de  $C$  qui permet d'obtenir une erreur globale (sur l'ensemble d'itérations) minimale est considérée la meilleure.

## 5.2 Points dans les quadrants $(x, y > 0)$ et $(x > 0, y < 0)$

On génère les points selon la procédure  $m = 2$  dans *generatedata.m*. Les points de la première classe sont dans le quadrat  $(x, y > 0)$  et ceux de la seconde classe sont dans le quadrat  $(x > 0, y < 0)$ . Voir le fichier *test1.mat* dans le dossier *test*. La meilleure valeur, au niveau du nombre d'erreurs, de  $C$ , choisie par validation croisée dans l'intervalle  $[1, 10]$ , est 1 (les autres valeurs de  $C$  de 2 à 10 donnent le même nombre d'erreurs pour cet ensemble de données).

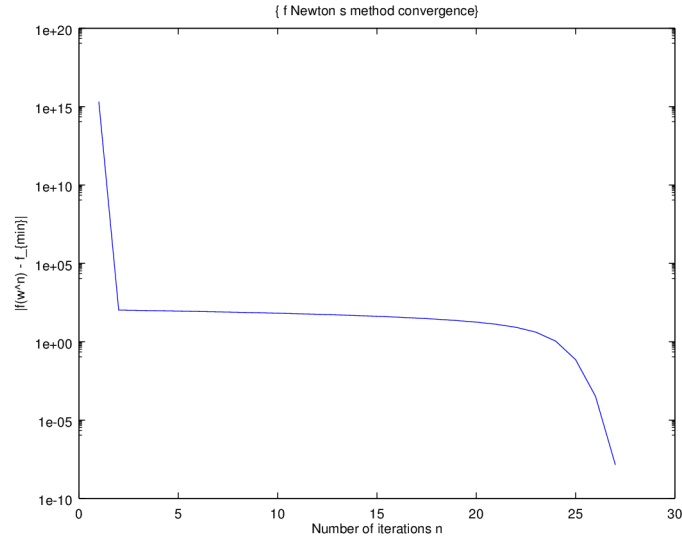


Figure 2: Ensemble de données 1 (échelle semi-log)

### 5.2.1 Pour $C = 1, n = 10, d = 40$

Table 2: Matrice de confusion pour l'ensemble de données 1

RÉALITÉ/PRÉDICTION	CLASSE 1	CLASSE 2
CLASSE 1	2	0
CLASSE 2	0	1

## 5.3 Points centrés réduits générés à partir de deux fonctions gaussiennes

On génère les points selon la procédure  $m = 0$  dans *generatedata.m* avec  $sep = 10$ . On tire les coordonnées en utilisant la fonction *randn*, qui retourne des éléments centrés réduits générés par une Gaussienne, auxquels on retire ou ajoute 10. Voir le fichier *test2.mat* dans le dossier *test*.

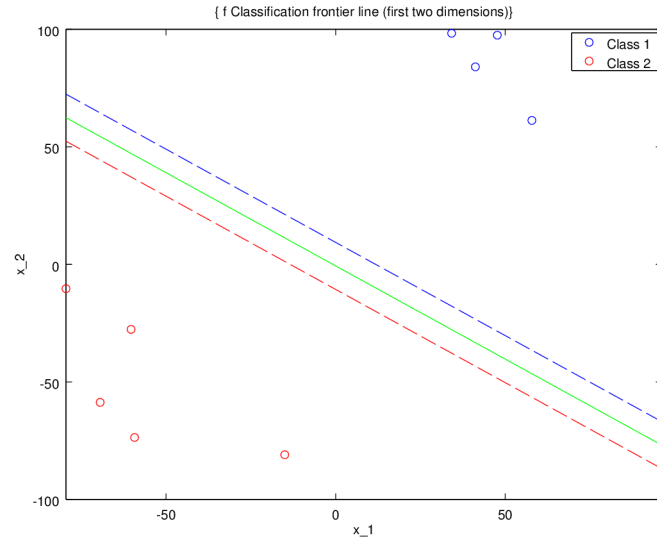


Figure 3: Ensemble de données 1 (Les lignes rouge et bleue en pointillés sont les droites  $f(x) = 10$  et  $f(x) = -10$ )

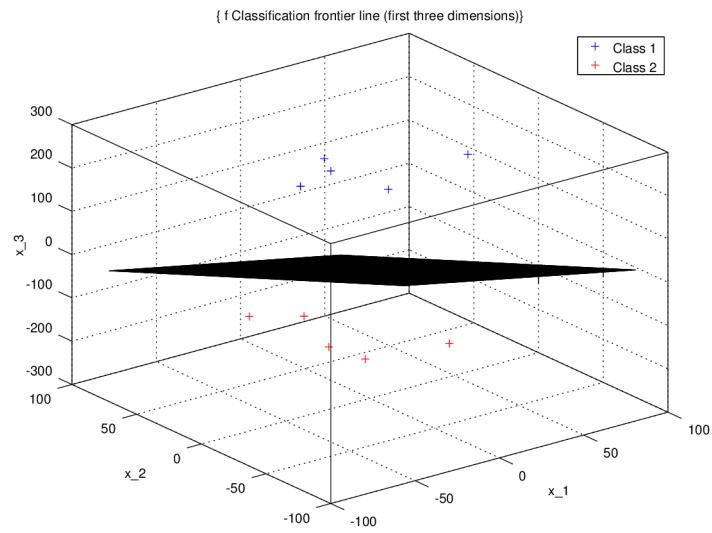


Figure 4: Ensemble de données 1



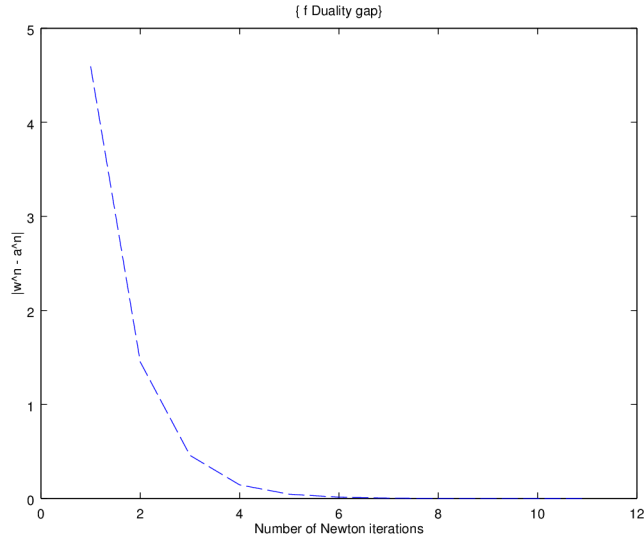


Figure 5: *Duality gap* : ensemble de données 1

#### 5.3.1 Pour $C = 5, n = 150, d = 200$

Table 3: Matrice de confusion pour l'ensemble de données 2

RÉALITÉ/PRÉDICTION	CLASSE 1	CLASSE 2
CLASSE 1	23	0
CLASSE 2	0	27

### 5.4 Points centrés réduits générés avec des fonctions gaussiennes

On utilise la procédure  $m = 0$  dans *generatedata.m* avec  $sep = 100$ . Voir le fichier *test3.mat* dans le dossier *test*.

#### 5.4.1 Pour $C = 5, n = 150, d = 200$

Table 4: Matrice de confusion pour l'ensemble de données 3

RÉALITÉ/PRÉDICTION	CLASSE 1	CLASSE 2
CLASSE 1	28	0
CLASSE 2	0	22

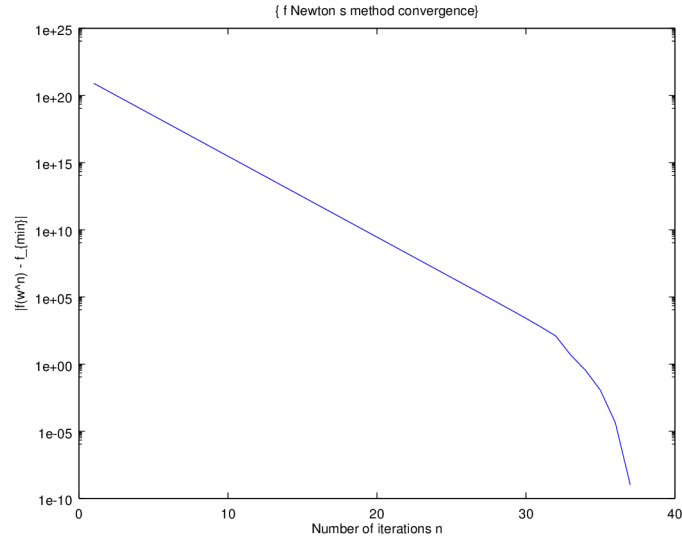


Figure 6: Ensemble de données 2 (échelle semi-log)

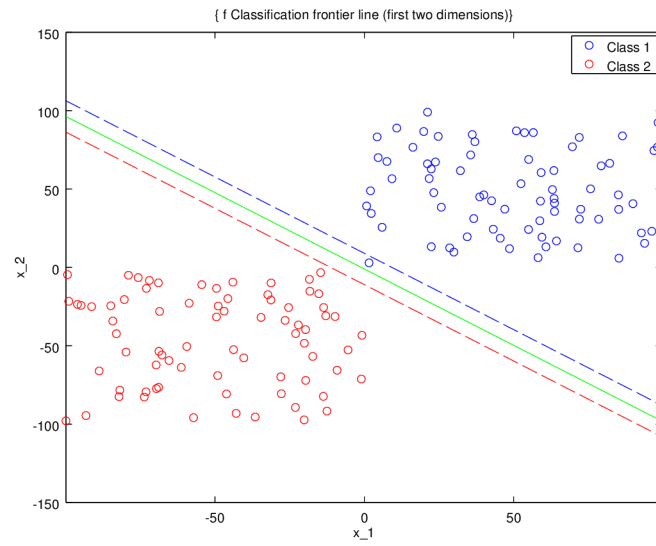


Figure 7: Ensemble de données 2 (Les lignes rouge et bleue en pointillés sont les droites  $f(x) = 10$  et  $f(x) = -10$ )

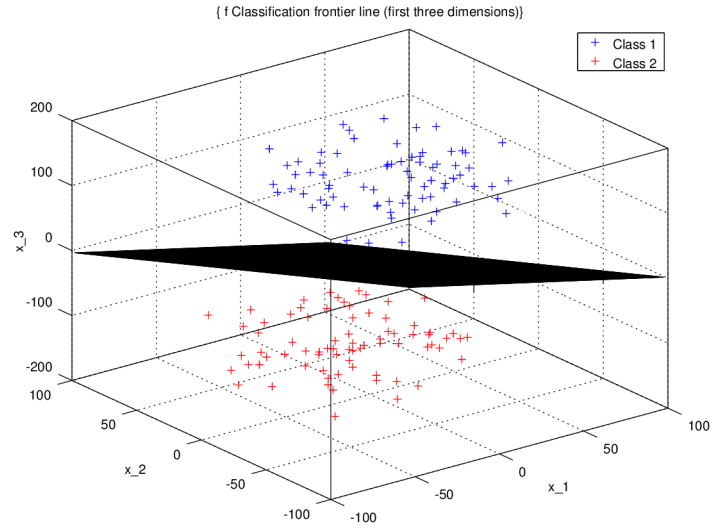


Figure 8: Ensemble de données 2

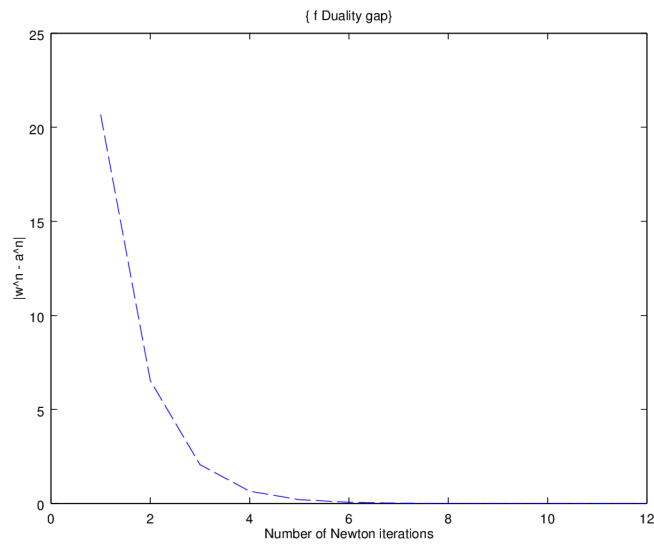


Figure 9: *Duality gap* : ensemble de données 2

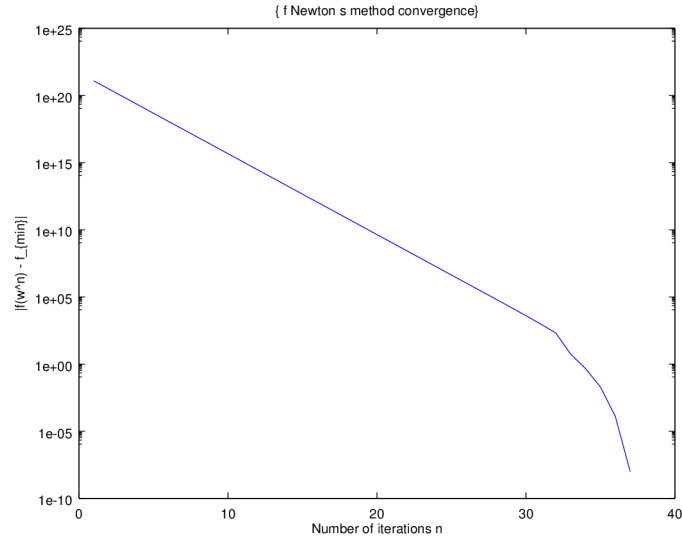


Figure 10: Ensemble de données 3 (échelle semi-log)

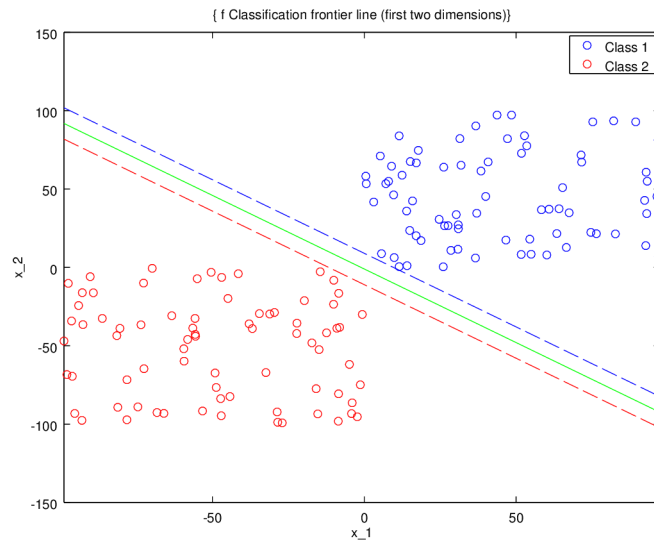


Figure 11: Ensemble de données 3 (Les lignes rouge et bleue en pointillés sont les droites  $f(x) = 10$  et  $f(x) = -10$ )

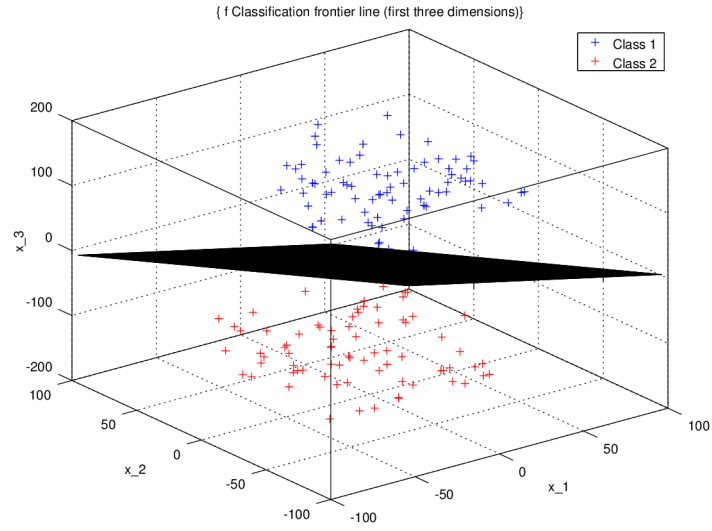


Figure 12: Ensemble de données 3

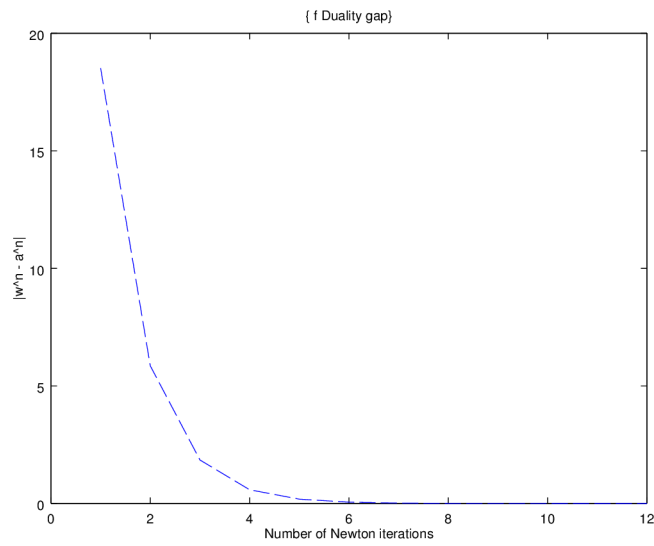


Figure 13: *Duality gap* : ensemble de données 3

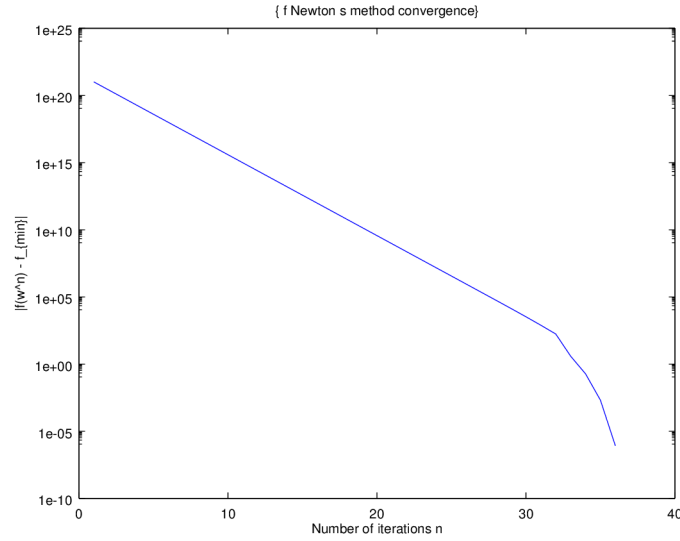


Figure 14: Ensemble de données 4 (échelle semi-log)

## 5.5 Points centrés réduits générés avec des fonctions gaussiennes

On utilise la procédure  $m = 0$  dans *generatedata.m* avec  $sep = 0$ . Voir le fichier *test4.mat* dans le dossier *test*.

### 5.5.1 Pour $C = 5, n = 150, d = 200$

Table 5: Matrice de confusion pour l'ensemble de données 4

RÉALITÉ/PRÉDICTION	CLASSE 1	CLASSE 2
CLASSE 1	28	0
CLASSE 2	0	22

## 5.6 Points générés avec des fonctions gaussiennes

On utilise la procédure  $m = 1$  dans *generatedata.m* avec les paramètres par défaut. Voir le fichier *test5.mat* dans le dossier *test*.

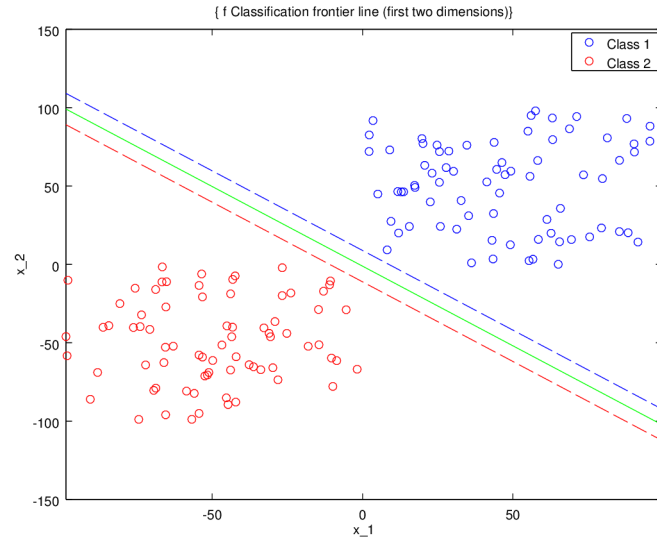


Figure 15: Ensemble de données 4 (Les lignes rouge et bleue en pointillés sont les droites  $f(x) = 10$  et  $f(x) = -10$ )

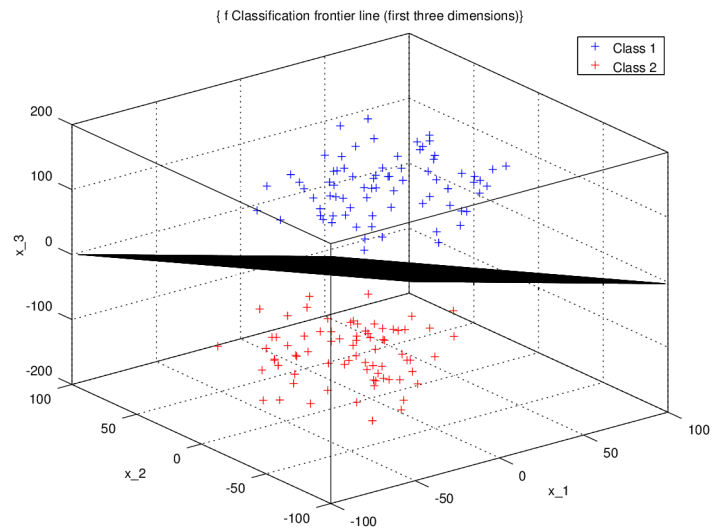


Figure 16: Ensemble de données 4

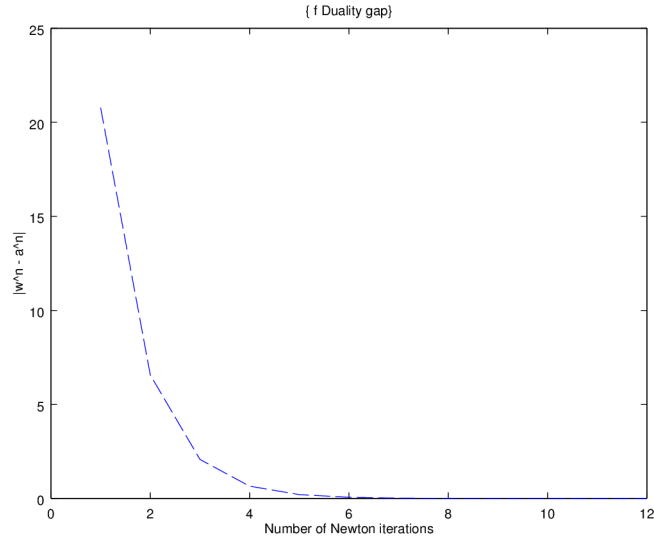


Figure 17: *Duality gap* : ensemble de données 4

#### 5.6.1 Pour $C = 5, n = 150, d = 200$

Table 6: Matrice de confusion pour l'ensemble de données 5

RÉALITÉ/PRÉDICTION	CLASSE 1	CLASSE 2
CLASSE 1	9	1
CLASSE 2	17	23



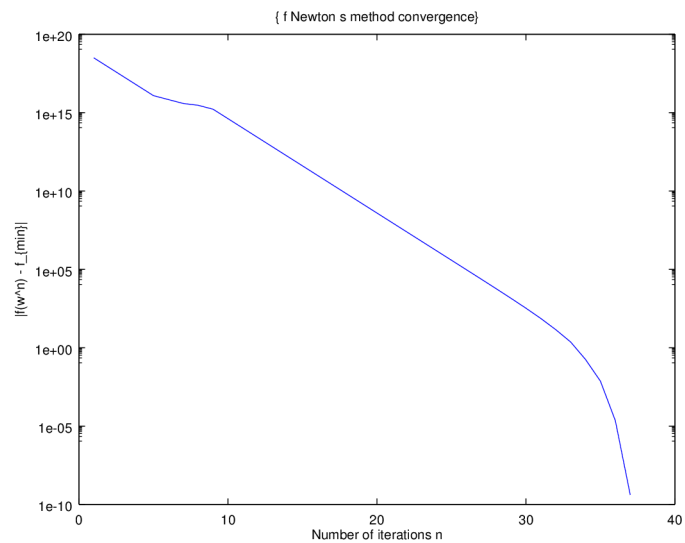


Figure 18: Ensemble de données 5 (échelle semi-log)

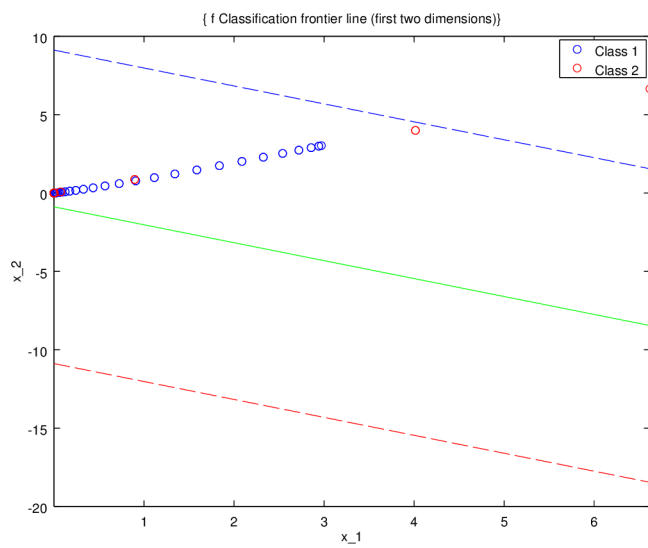


Figure 19: Ensemble de données 5 (Les lignes rouge et bleue en pointillés sont les droites  $f(x) = 10$  et  $f(x) = -10$ )

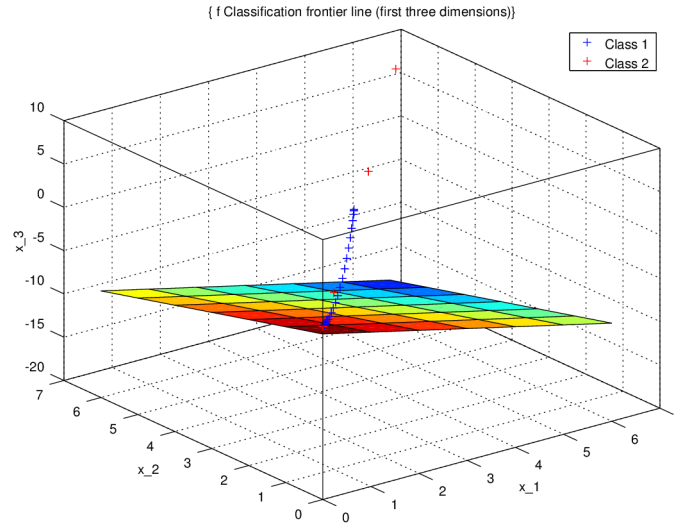


Figure 20: Ensemble de données 5

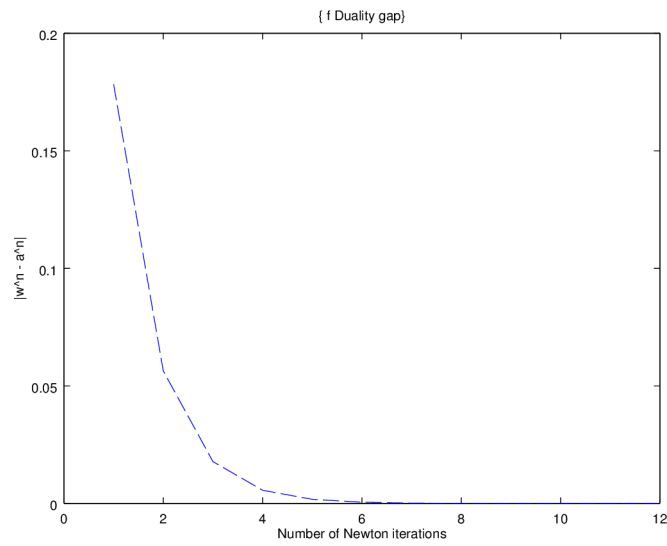


Figure 21: *Duality gap* : ensemble de données 5