Projet Optimisation : Support Vector Machine

K. Kamtue & Cl. Réda

ENS Cachan

12 janvier 2017

Sujet

Support Machine Vector

Objectif

Faire de l'apprentissage supervisé

Sujet

Support Machine Vector

Objectif

Faire de l'apprentissage supervisé

- Appliqué à la **classification binaire** ($y_i \in \{1, -1\}$);

Sujet

Support Machine Vector

Objectif

Faire de l'apprentissage supervisé

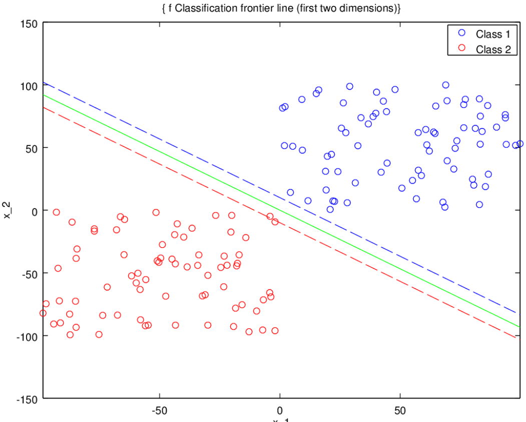
- Appliqué à la **classification binaire** ($y_i \in \{1, -1\}$);
- Recherche d'une **frontière linéaire** $f : x \to \omega^T x$ vérifiant :

$$\forall i, y_i = -1 \Rightarrow f(x_i) < 0$$
$$\forall i, y_i = 1 \Rightarrow f(x_i) > 0$$
$$\Leftrightarrow \forall i, y_i \times f(x_i) > 0 \ (1)$$

Sujet

Support Machine Vector

Figure: Exemple avec deux classes (rouge et bleue)



Le problème d'optimisation

Recherche du problème d'optimisation

Le problème d'optimisation (naïvement)

γ : distance entre les droites $f(x) = 1$ et $f(x) = -1$ (**marge**).

Le problème d'optimisation

Recherche du problème d'optimisation

Le problème d'optimisation (naïvement)

γ : distance entre les droites $f(x) = 1$ et $f(x) = -1$ (**marge**).

$$max_w \ \gamma = \frac{2}{||w||}$$
$$\text{avec } \forall i, y_i \times f(x_i) > 0$$

Le problème d'optimisation

Recherche du problème d'optimisation

Le problème d'optimisation (naïvement)

γ : distance entre les droites $f(x) = 1$ et $f(x) = -1$ (**marge**).

$$max_w \ \gamma = \frac{2}{\|w\|}$$
$$\text{avec } \forall i, y_i \times f(x_i) > 0$$

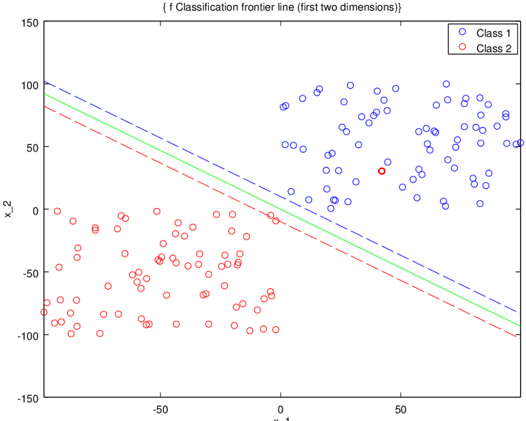
$$\Leftrightarrow min_w \ \frac{1}{2}\|w\|^2$$
$$\text{avec } \forall i, y_i \times f(x_i) > 0$$

Attention : si l'ensemble n'est pas séparable !

Le problème d'optimisation

Recherche du problème d'optimisation

Figure: Exemple avec deux classes (rouge et bleue)



Le problème d'optimisation

Adaptation au cas non séparable

Soit $z_i = max(0, 1 - y_i \times f(x_i))$ (perte de **Hinge**).

Le problème d'optimisation

Adaptation au cas non séparable

Soit $z_i = max(0, 1 - y_i \times f(x_i))$ (perte de **Hinge**).

En rendant le problème toujours faisable et convexe

Pénaliser les erreurs de classification avec les $(z_i)_i$ et C :

$$min_{w,z} \ \frac{1}{2}\|w\|^2 + C \sum_{i \leq m} z_i$$
$$\text{avec } \forall i, z_i \geq 0$$
$$\forall i, y_i \times (\omega^T x_i) \geq 1 - z_i$$

Implémentation

Résolution du problème d'optimisation

- Utilisation de la **méthode de Newton** pour trouver ω :

Rappel : Mise à jour du vecteur x cherché avec la méthode de Newton

$$x_{n+1} \leftarrow x_n + s \times \nabla^2 obj(x_n)^{-1} \nabla obj(x_n)$$

(ici, en cherchant s, taille du pas, par **backtracking line search**)

Implémentation
Résolution du problème d'optimisation

- Utilisation de la **méthode de Newton** pour trouver ω :

Rappel : Mise à jour du vecteur x cherché avec la méthode de Newton

$$x_{n+1} \leftarrow x_n + s \times \nabla^2 obj(x_n)^{-1} \nabla obj(x_n)$$

(ici, en cherchant s, taille du pas, par **backtracking line search**)

- Utiliser la **méthode de la barrière logarithmique**.

Implémentation

Résolution du problème d'optimisation

- Utilisation de la **méthode de Newton** pour trouver ω :

Rappel : Mise à jour du vecteur x cherché avec la méthode de Newton

$$x_{n+1} \leftarrow x_n + s \times \nabla^2 obj(x_n)^{-1} \nabla obj(x_n)$$

(ici, en cherchant s, taille du pas, par **backtracking line search**)

- Utiliser la **méthode de la barrière logarithmique**.
- Rendre le problème indépendant de la **dimension**;

Implémentation

Indépendance en la dimension des points : **problème dual**

Après calcul du lagrangien et minimisation en ω :

Implémentation

Indépendance en la dimension des points : **problème dual**

Après calcul du lagrangien et minimisation en ω :

Problème dual

$$max_{\lambda \in \mathbb{R}^{+m}} - \frac{1}{2} \| \sum_i \lambda_i y_i x_i \|_2^2 + \mathbf{1}^T \lambda$$

avec $\forall i, 0 \leq \lambda_i \leq C$

(par les conditions de KKT)

Implémentation

Indépendance en la dimension des points : **problème dual**

Après calcul du lagrangien et minimisation en ω :

Problème dual

$$max_{\lambda \in \mathbb{R}^{+m}} - \frac{1}{2} \| \sum_i \lambda_i y_i x_i \|_2^2 + \mathbf{1}^T \lambda$$

avec $\forall i, 0 \leq \lambda_i \leq C$

(par les conditions de KKT)

Obtenir la solution du primal à partir de celle du dual

$$\omega^* = \sum_i \lambda_i^* y_i x_i$$

Implémentation

Rendre le problème indépendant de la dimension

Utilisation de l'*astuce du noyau* :

Problème dual

Soit $K = X^T X$ (noyau linéaire). Alors :

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

Implémentation

Supprimer les contraintes d'inégalité

Utilisation de la **méthode de la barrière logarithmique** :

Implémentation

Supprimer les contraintes d'inégalité

Utilisation de la **méthode de la barrière logarithmique** :

Fonction barrière pour éliminer les contraintes d'inégalité

$$\Phi(\lambda) = \sum_i (-log(C - \lambda_i) - log(\lambda_i))$$
$$= -\sum_i log((C - \lambda_i)\lambda_i)$$

Implémentation

Supprimer les contraintes d'inégalité

Utilisation de la **méthode de la barrière logarithmique** :

Fonction barrière pour éliminer les contraintes d'inégalité

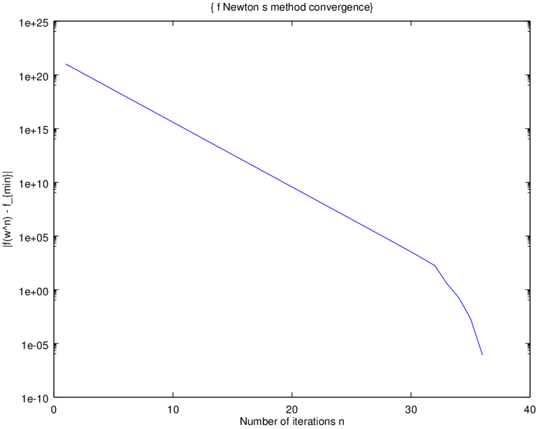
$$\begin{aligned}\Phi(\lambda) &= \sum_i(-log(C-\lambda_i) - log(\lambda_i)) \\ &= -\sum_i log((C-\lambda_i)\lambda_i)\end{aligned}$$

Problème d'optimisation final

$$max\ -\frac{1}{2}\lambda^T diag(y) K diag(y)\lambda + \mathbf{1}^T\lambda + \Phi(\lambda)$$

Détails de l'implémentation

Tracé de la convergence de la méthode de Newton



Détails de l'implémentation

Dépendance en la taille de l'échantillon

Figure: Comparaison des temps de calcul selon n et d
Ensemble | C | d | n | Nb d'itérations | temps

1	5	40000	10	11	0,315
1	5	40	100	12	0,715
1	5	40	1000	?	¿ 10

Détails de l'implémentation

Accélération de la convergence quand C augmente

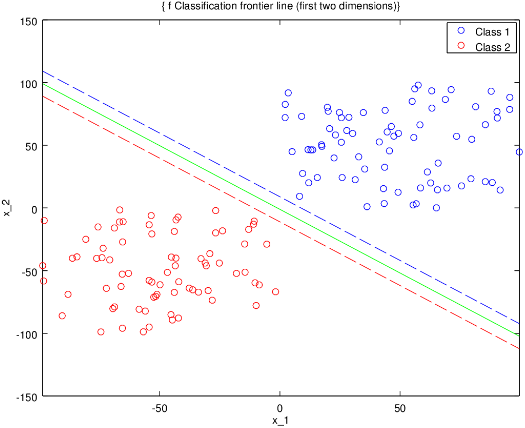
Figure: Evolution du temps de calcul en fonction de C

Test	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 (*)

Tracé de la frontière de classification

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

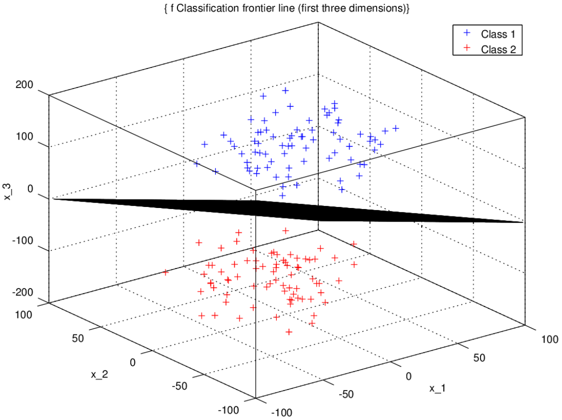
Points centrés réduits avec des fonctions gaussiennes (2D) :



Tracé de la frontière de classification

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

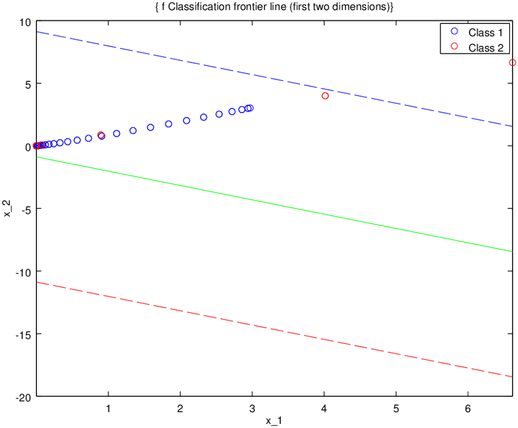
Points centrés réduits avec des fonctions gaussiennes (3D) :



Tracé de la frontière de classification

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

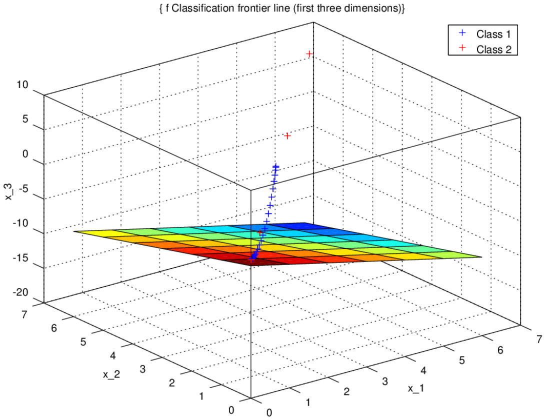
Génération avec des fonctions gaussiennes (2D) :



Tracé de la frontière de classification

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

Génération avec des fonctions gaussiennes (3D) :



Extensions

Ajouts au projet

- **Validation croisée** (choix de la meilleure valeur de C);

Extensions

Ajouts au projet

- **Validation croisée** (choix de la meilleure valeur de C);
- Implémentation de **Coordinate Descent**;

Extensions

Ajouts au projet

- **Validation croisée** (choix de la meilleure valeur de C);
- Implémentation de **Coordinate Descent**;
- Implémentation de **ACCPM**;

Démontration du SVM