Optimization Project: Support Vector Machine

K. Kamtue & Cl. Réda

ENS Cachan

January 12th, 2017

Project

Support Machine Vector

Objective

Classify data

Project

Support Machine Vector

Objective

Classify data

- Applied to **binary classification** ($y_i \in \{1, -1\}$);

Project

Support Machine Vector

Objective

Classify data

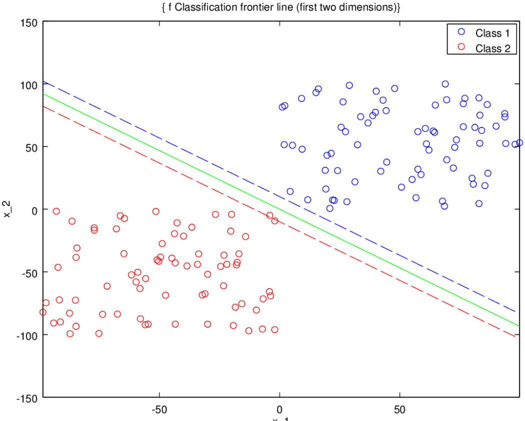
- Applied to **binary classification** ($y_i \in \{1, -1\}$);
- Looking for a **hyperplane** $f : x \rightarrow \omega^T x$ such as:

$$\forall i, f(x_i) = \begin{cases} < 0 & \text{si y=-1} \\ > 0 & \text{si y=1} \end{cases} \Leftrightarrow \forall i, y_i \times f(x_i) > 0 \qquad (1)$$

Project

Support Machine Vector

Figure: Example with two classes (**red** and **blue**)



Optimization problem

Looking for the optimization problem

Naive optimization problem

γ: distance between the lines $f(x) = 1$ and $f(x) = -1$ (**margin**).

Optimization problem

Looking for the optimization problem

Naive optimization problem

γ: distance between the lines $f(x) = 1$ and $f(x) = -1$ (**margin**).

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

Optimization problem

Looking for the optimization problem

Naive optimization problem

γ: distance between the lines $f(x) = 1$ and $f(x) = -1$ (**margin**).

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

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

Beware: if the data set is not linearly separable!

Optimization

Looking for the optimization problem

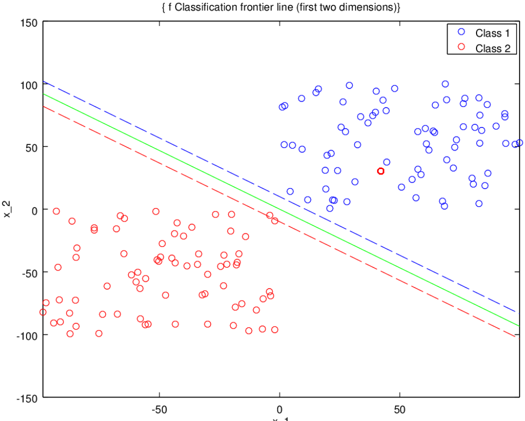


Figure: Example with two classes (red and blue)

Optimization problem

Adapting the problem to **non-separable sets**

Let z_i be $max(0, 1 - y_i \times f(x_i))$ (**Hinge loss**).

Optimization problem

Adapting the problem to **non-separable sets**

Let z_i be $max(0, 1 - y_i \times f(x_i))$ (**Hinge loss**).

Having the problem convex and always feasible

Penalty for **classification errors** with **slack variables** $(z_i)_i$ and C :

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

Implementation

Solving the optimization problem

- Using **Newton's method** to find ω :

Reminder: Update of ω with **Newton's method**

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

(finding **step size** value s by **backtracking line search**)

Implementation

Solving the optimization problem

- Using **Newton's method** to find ω :

Reminder: Update of ω with **Newton's method**

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

(finding **step size** value s by **backtracking line search**)

- Make the problem independant of **dimension**;

Implementation

Solving the optimization problem

- Using **Newton's method** to find ω :

Reminder: Update of ω with **Newton's method**

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

(finding **step size** value s by **backtracking line search**)

- Make the problem independant of **dimension**;
- Using **logarithmic barrier method**.

Implementation

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

Après calcul du lagrangien et minimisation en ω :

Implementation

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)

Implementation

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

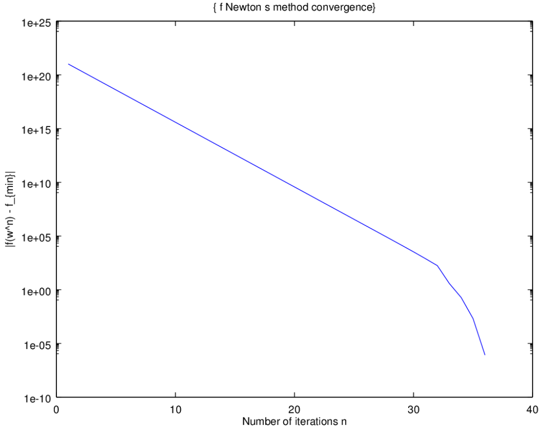
$$\Phi(\lambda) = \sum_i (-log(C - \lambda_i) - log(\lambda_i)) \\ = -\sum_i log((C - \lambda_i)\lambda_i)$$

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

Ensemble	C	d	n	Nb d'itération	temps
1	5	40000	10	11	0.315
1	5	40	100	12	0.715
1	5	40	1000	very large	>1000

Détails de l'implémentation

Accélération de la convergence quand C augmente

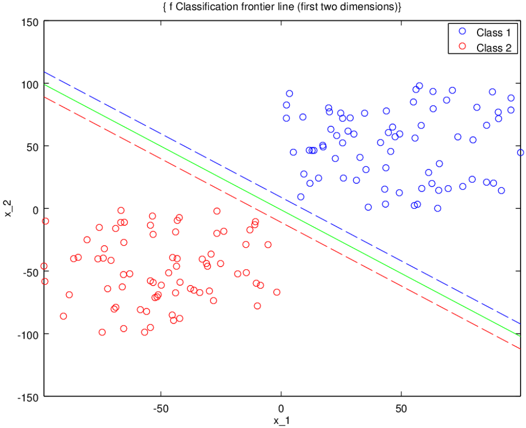
Figure: Evolution du temps de calcul en fonction de C

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

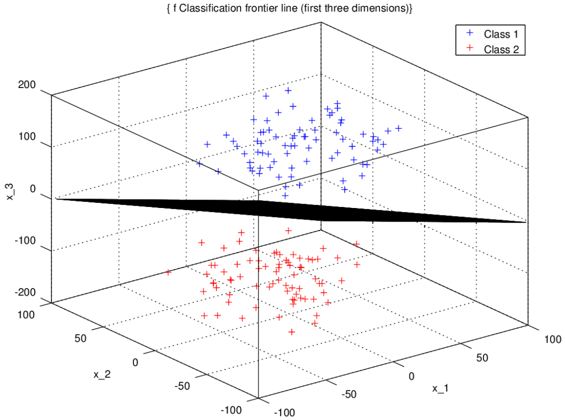
Points centrés réduits avec des fonctions gaussiennes (2D) :



Tracé de la frontière de classification

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

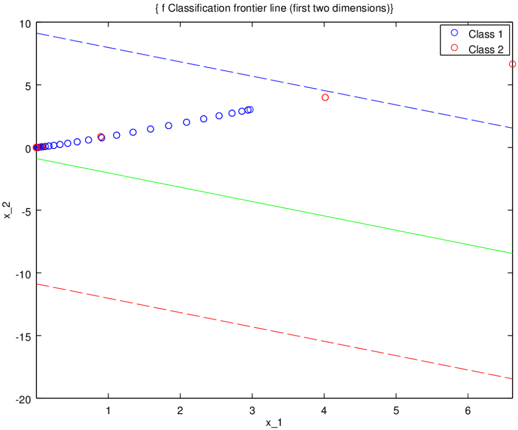
Points centrés réduits avec des fonctions gaussiennes (3D) :



Tracé de la frontière de classification

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

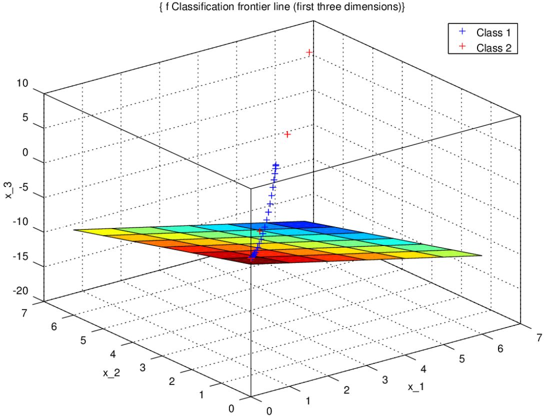
Génération avec des fonctions gaussiennes (2D) :



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**;

Démonstration du SVM