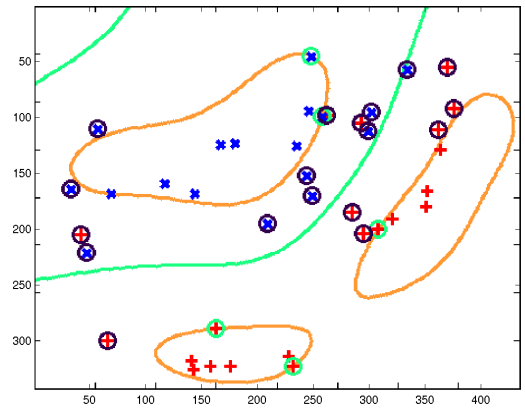
Séparateurs à Vaste Marge
Une méthode à noyau pour la discrimination

Gaëlle Loosli
gaelle@loosli.fr

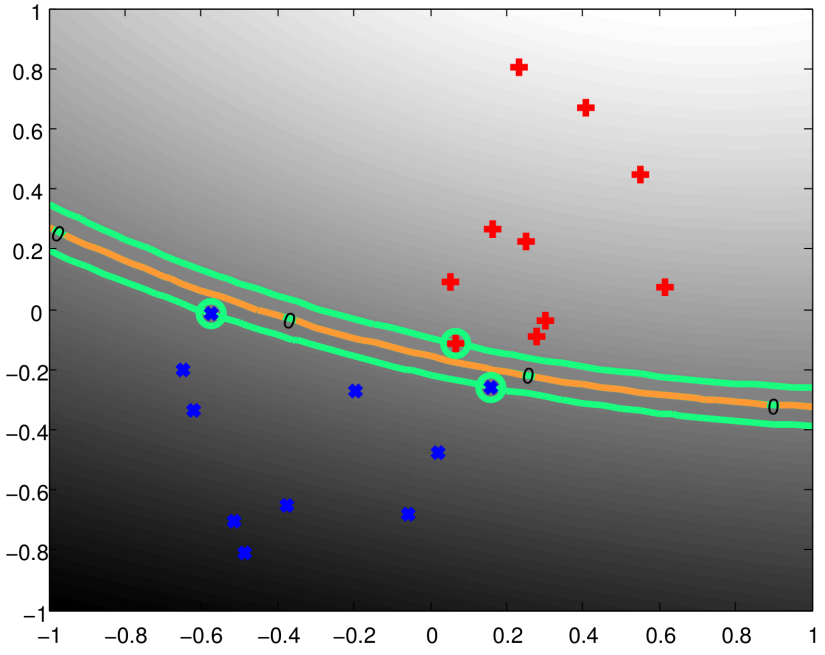
GMM3, Polytech'Clermont

janvier 2010



Pourquoi parler des SVMs ?

Le SVM est une méthode qui permet de tracer de manière optimale une frontière entre deux classes de points :



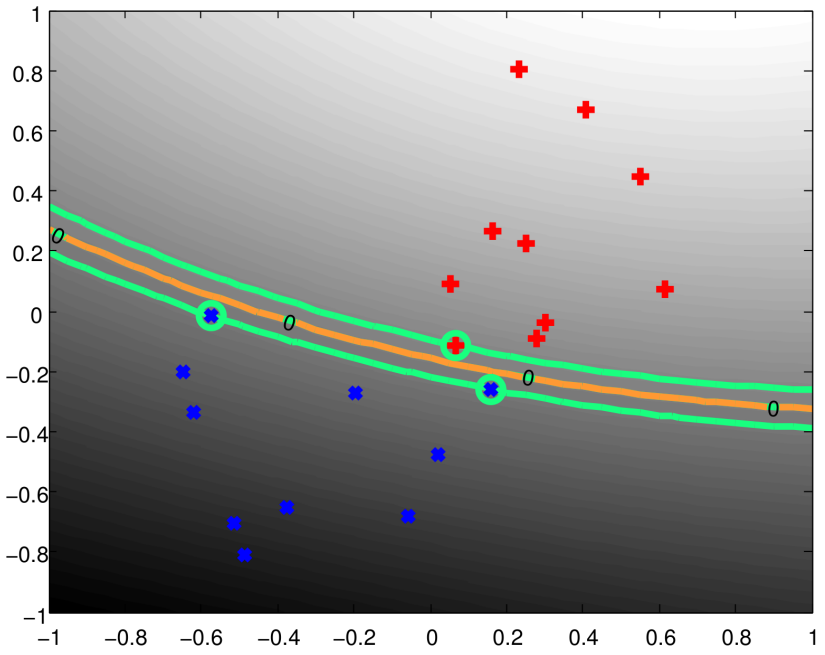
Question binaire :

- Cette image est elle un arbre ou un oiseau ?
- Cette pièce est-elle défectueuse ?
- L'état de mon système est-il viable ?

Pourquoi parler des SVMs ?

Le SVM est une méthode qui permet de tracer de manière optimale une frontière entre deux classes de points :

- Solution unique
- Solution parcimonieuse
- Solution non-linéaire
- Programmation simple
- Peu d'hyper-paramètres



Les types d'apprentissage - non exhaustif

En fonction du type de sortie attendue

- prédiction de valeur : la régression
- prédiction de classe : la discrimination
- une commande : l'apprentissage par renforcement

En fonction de l'information disponible

- toutes les étiquettes : l'apprentissage supervisé
- aucune étiquette : l'apprentissage non supervisé
- une partie des étiquettes : l'apprentissage semi-supervisé (transductif)
- des étiquettes à la demande : l'apprentissage actif

Les types d'apprentissage - non exhaustif

En fonction du type de sortie attendue

- prédiction de valeur : la régression
- prédiction de classe : la discrimination
- une commande : l'apprentissage par renforcement

En fonction de l'information disponible

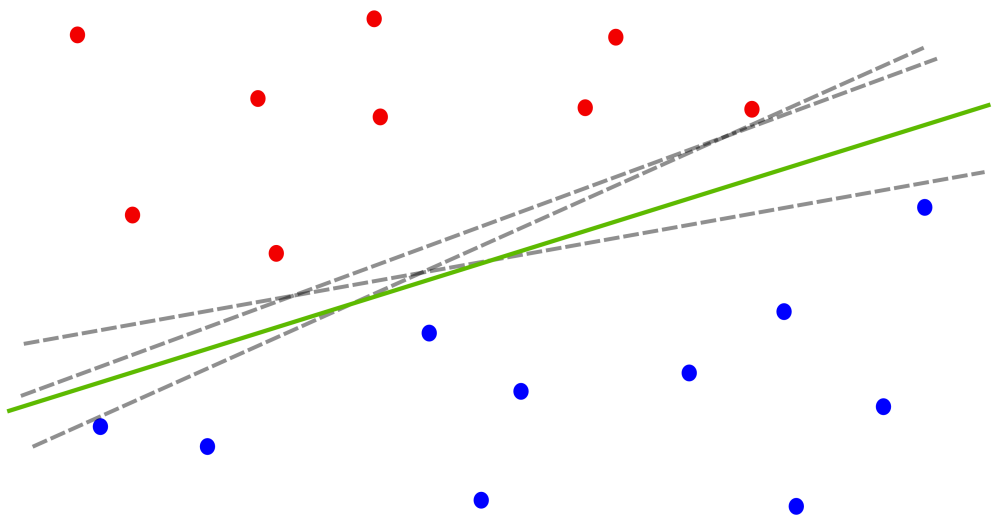
- toutes les étiquettes : l'apprentissage supervisé
- aucune étiquette : l'apprentissage non supervisé
- une partie des étiquettes : l'apprentissage semi-supervisé (transductif)
- des étiquettes à la demande : l'apprentissage actif

Apport des SVMs

Minimisation quadratique sous contraintes :

$$\begin{cases} \min\limits_{f,b} \dfrac{1}{2}\|f\|^2 \\ y_i\big(f(x_i)+b\big) \geq 1 \quad \forall i \in [1,...,m] \end{cases}$$

- Solution unique
- Solution parcimonieuse : remise en contexte
- Solution non-linéaire
- Programmation simple
- Peu d'hyper-paramètres



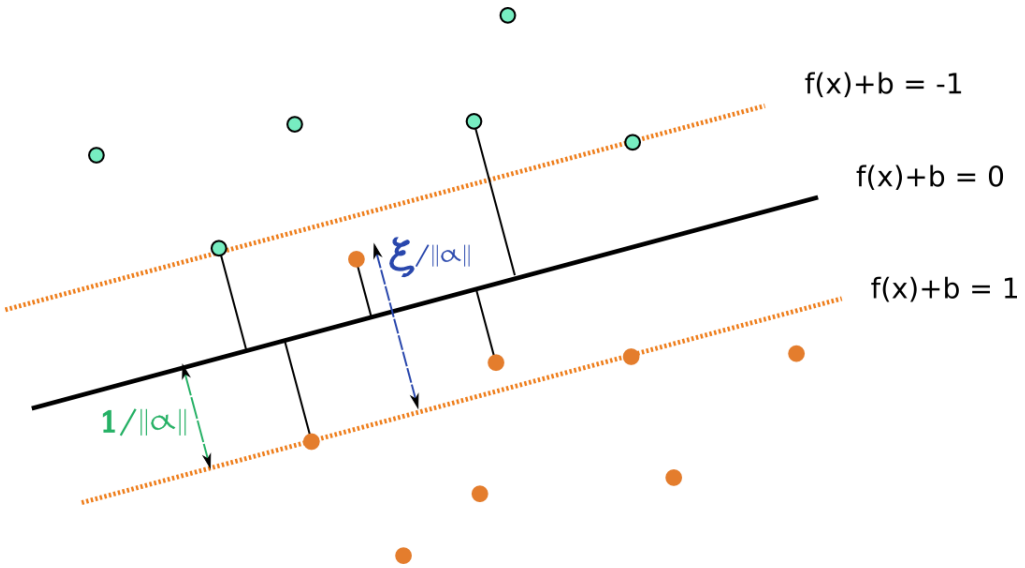
Apport des SVMs

La solution se présente sous la forme d'une combinaison linéaire des points d'apprentissage. Un grand nombre des coefficients sont nuls : c'est la parcimonie.

$$D(x) = sign(f(x) + b)$$

$$f(.) = \sum_{i=1}^{n} \alpha_i \langle x_i, . \rangle$$

- Solution unique
- Solution parcimonieuse
- Solution non-linéaire
- Programmation simple
- Peu d'hyper-paramètres



Apprentissage supervisé : ce qui coûte cher

La plupart des algorithmes d'apprentissage supervisé reviennent à résoudre un système linéaire de taille $n \times n$:

$$Ax = b$$

$$\begin{bmatrix} & A & \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} b \end{bmatrix}$$

avec A une matrice carrée de taille n, contenant l'information issue des données d'apprentissage, b un vecteur colonne de taille n contenant les étiquettes, et x le vecteur des inconnues.

Apprentissage supervisé : résolution directe

$$\begin{bmatrix} & A & \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} b \end{bmatrix}$$

A est souvent symétrique et définie positive : utilisation de la décomposition de Cholesky :

$$\begin{array}{lcl} A = LL' & \Rightarrow & \mathcal{O}(\frac{1}{3}n^3) \\ Lz = b & \Rightarrow & \mathcal{O}(n^2) \\ L'x = z & \Rightarrow & \mathcal{O}(n^2) \end{array}$$

Apprentissage supervisé : résolution directe

$$\begin{bmatrix} & A & \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} b \end{bmatrix}$$

A est souvent symétrique et définie positive : utilisation de la décomposition de Cholesky :

$$\begin{array}{lcl} A = LL' & \Rightarrow & \mathcal{O}(\frac{1}{3}n^3) \\ Lz = b & \Rightarrow & \mathcal{O}(n^2) \\ L'x = z & \Rightarrow & \mathcal{O}(n^2) \end{array}$$

Ne convient qu'à des systèmes de taille restreinte.

Apprentissage supervisé : résolution itérative

$$\begin{bmatrix} & A & \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} b \end{bmatrix}$$

On initialise x_0 (aléatoirement par exemple).
Puis on met à jour selon la règle suivante :

$$x_{i+1} = x_i + \delta d \quad \Rightarrow \quad \mathcal{O}(n^2)$$

avec δ un pas et d une direction.

- Le pas peut-être fixe ou variable
- La direction est définie selon le gradient (du premier ou second ordre).

Ces méthodes permettent une résolution exacte en n étapes ($\Rightarrow \mathcal{O}(n^3)$) mais donnent une bonne approximations en quelques étapes ($\Rightarrow \mathcal{O}(kn^2)$).

Permet d'envisager des systèmes plus grand que la résolution directe, mais limité tout de même et au prix de la précision des résultats.

Apprentissage supervisé : utilisons la parcimonie

$$\begin{bmatrix} & A & \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} b \end{bmatrix}$$

Principe : forcer un maximum de valeurs de x à être nulles.

$$\begin{bmatrix} - & - & - \\ - & - & - \\ - & - & A_p \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ x_p \end{bmatrix} = \begin{bmatrix} - \\ - \\ b_p \end{bmatrix}$$

Apprentissage supervisé : utilisons la parcimonie

$$\begin{bmatrix} & A & \end{bmatrix} \begin{bmatrix} x \end{bmatrix} = \begin{bmatrix} b \end{bmatrix}$$

Principe : forcer un maximum de valeurs de x à être nulles.

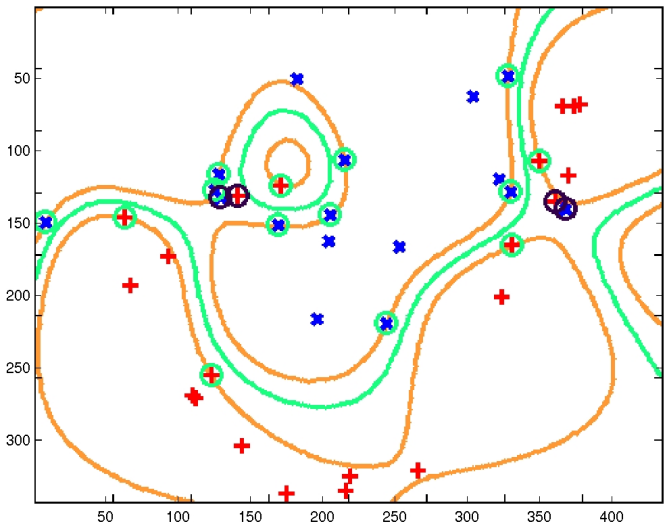
$$\begin{bmatrix} - & - & - \\ - & - & - \\ - & - & A_p \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ x_p \end{bmatrix} = \begin{bmatrix} - \\ - \\ b_p \end{bmatrix}$$

Sélectionner une fraction des points pour la solution permet de considérablement augmenter la dimension des problèmes à traiter.

Apport des SVMs

Les points d'apprentissage sont virtuellement projetés dans un espace de grande dimension dans lequel ils sont linéairement séparables : on utilise pour cela un noyau.

- Solution unique
- Solution parcimonieuse
- Solution non-linéaire dans l'espace de départ
- Programmation simple
- Peu d'hyper-paramètres



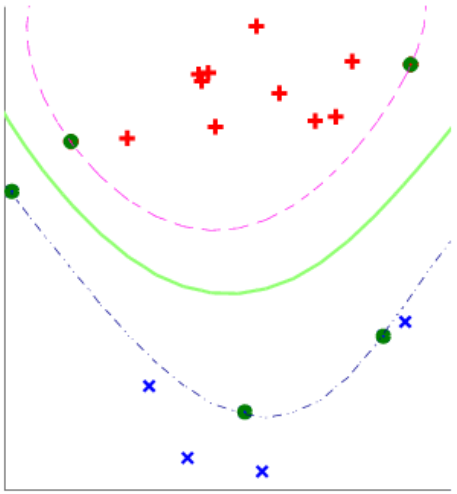
Les noyaux : SVMs non linéaires

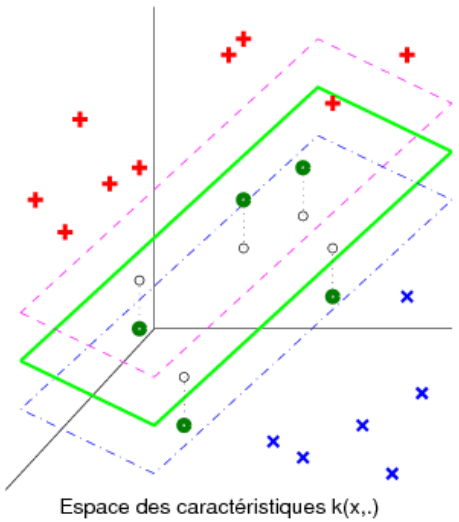
Utilité du noyau

Soit

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$$

avec $\phi(.)$ une fonction de projection.





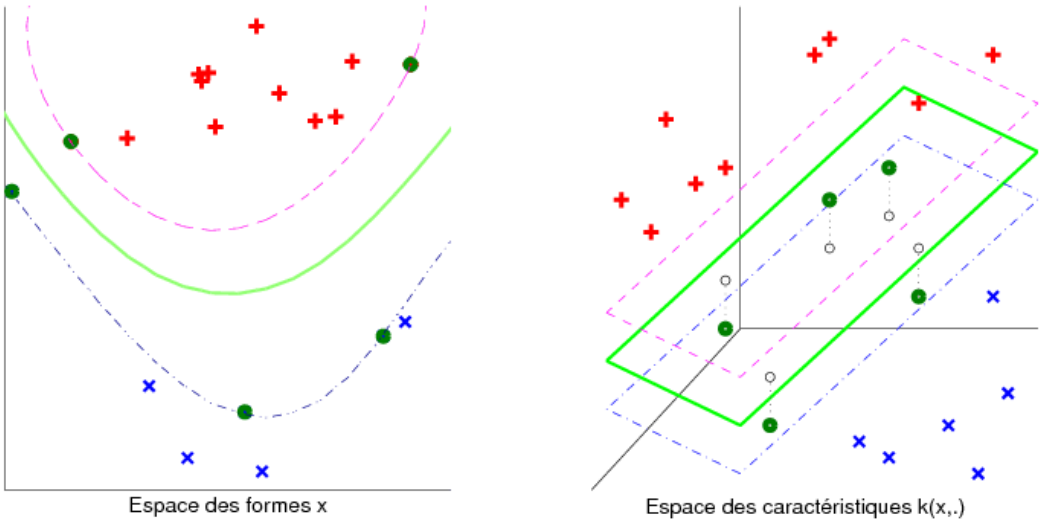
Les noyaux : SVMs non linéaires

Utilité du noyau

Soit

$$k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$$

avec $\phi(.)$ une fonction de projection.

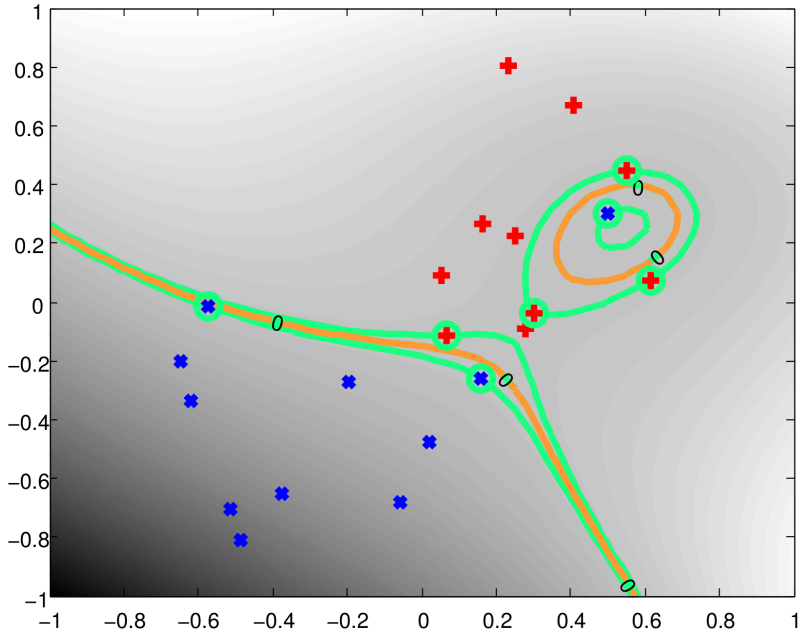


La bonne nouvelles : le théorème de représentation indique qu'il n'est pas nécessaire de connaître ϕ. On utilise une fonction noyau telle que : $k(x_i, x_j) = \exp \frac{-\|x_i - x_j\|}{2\sigma^2}$, le noyau Gaussien qui projette dans un espace de dimension infinie.

Apport des SVMs

Les seuls hyper-paramètres à régler pour utiliser les SVMs sont ceux liés au noyau et celui lié à la régularisation de la solution : il autorise ou non de faire des erreurs au cours de l'apprentissage (afin d'éviter le sur-apprentissage)

- Solution unique
- Solution parcimonieuse
- Solution non-linéaire
- Programmation simple
- Peu d'hyper-paramètres



Contraintes strictes ou non ?

Contraintes strictes

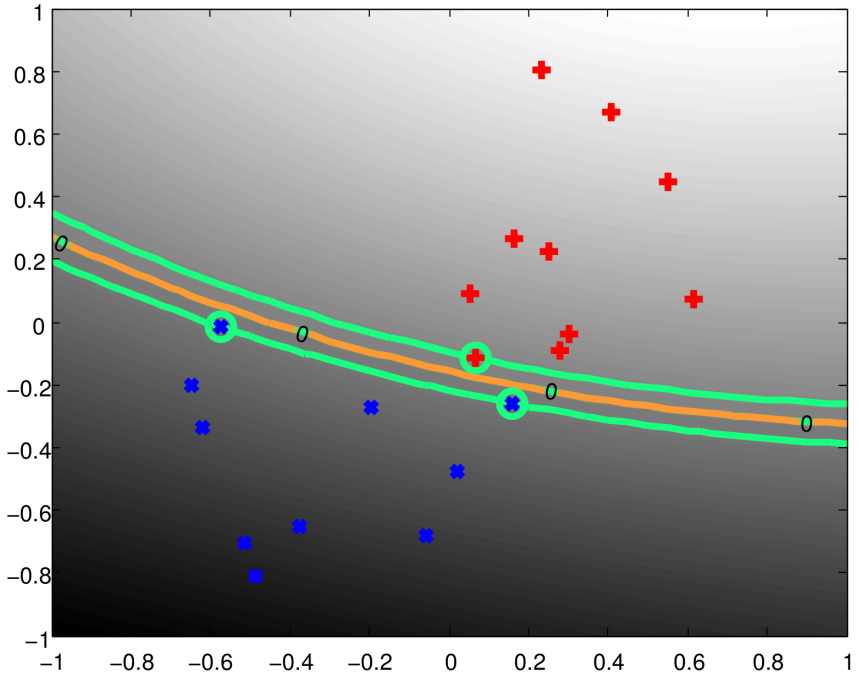
Problème primal :

$$\begin{cases} \min\limits_{f,b} \dfrac{1}{2}\|f\|^2 \\ y_i\big(f(x_i)+b\big) \geq 1 \quad \forall i \in [1,...,m] \end{cases}$$

Problème dual :

$$\begin{cases} \max\limits_{\boldsymbol{\alpha}} -\dfrac{1}{2}\boldsymbol{\alpha}^\top G\boldsymbol{\alpha} - \boldsymbol{\alpha}^\top \mathbf{1} \\ \boldsymbol{\alpha}^\top \mathbf{y} = 0 \\ 0 \leq \alpha_i \qquad\qquad \forall i \in [1,...,m] \end{cases}$$

avec $G(i,j) = \frac{1}{m} y_i y_j k(x_i, x_j)$.



Contraintes strictes ou non ?

Contraintes strictes

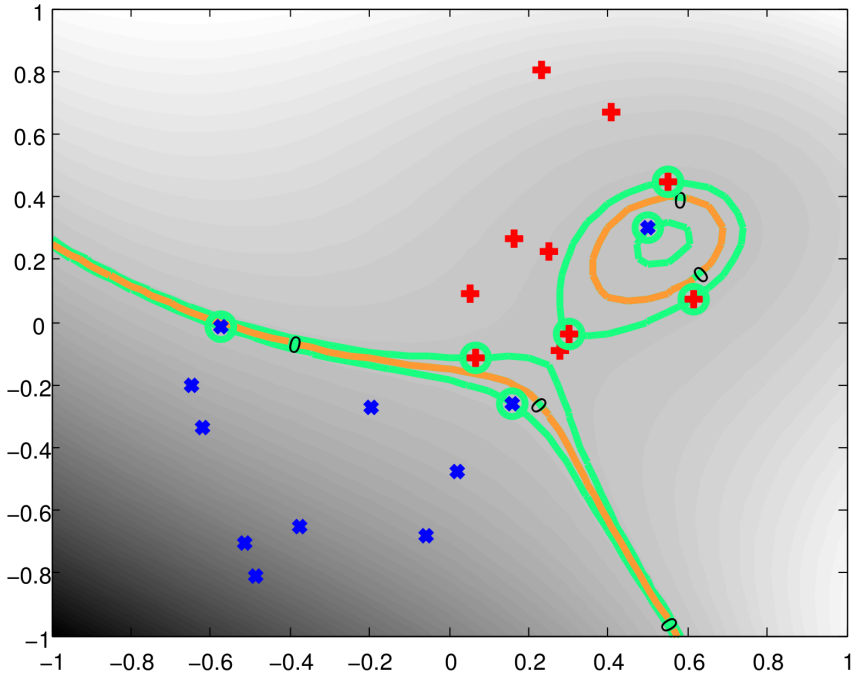
Problème primal :

$$\begin{cases} \min\limits_{f,b} \dfrac{1}{2}\|f\|^2 \\ y_i\big(f(x_i)+b\big) \geq 1 \qquad \forall i \in [1,...,m] \end{cases}$$

Problème dual :

$$\begin{cases} \max\limits_{\boldsymbol{\alpha}} -\dfrac{1}{2}\boldsymbol{\alpha}^\top G \boldsymbol{\alpha} - \boldsymbol{\alpha}^\top \mathbf{1} \\ \boldsymbol{\alpha}^\top \mathbf{y} = 0 \\ 0 \leq \alpha_i \qquad\qquad \forall i \in [1,...,m] \end{cases}$$

avec $G(i,j) = \frac{1}{m} y_i y_j k(x_i, x_j)$.



Contraintes strictes ou non ?

Contraintes douces

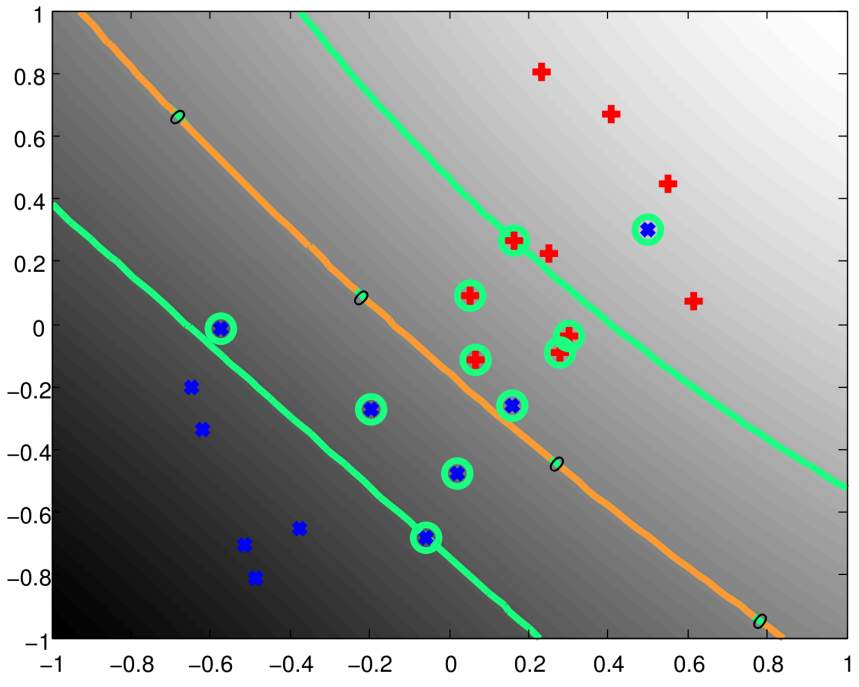
Problème primal :

$$\begin{cases} \min\limits_{f,b,\xi} \frac{1}{2}\|f\|^2 + C\sum\limits_{i=1}^{m}\xi_i \\ y_i\big(f(x_i)+b\big) \geq 1-\xi_i & \forall i \in [1,...,m] \\ \xi_i \geq 0 & \forall i \in [1,...,m] \end{cases}$$

Problème dual :

$$\begin{cases} \max\limits_{\boldsymbol{\alpha}} -\frac{1}{2}\boldsymbol{\alpha}^\top G\boldsymbol{\alpha} - \boldsymbol{\alpha}^\top \mathbf{1} \\ \boldsymbol{\alpha}^\top \mathbf{y} = 0 \\ 0 \leq \alpha_i \leq C & \forall i \in [1,...,m] \end{cases}$$

avec $G(i,j) = \frac{1}{m} y_i y_j k(x_i, x_j)$.



En pratique

Pour utiliser les SVMs, il faut :

- se munir d'un bon solver (libSVM, SVM^{light}, SimpleSVM, ...)
- choisir un noyau (Gaussien, polynomial, spécialisé pour des données structurées, ...)
- évaluer la possibilité d'erreurs d'étiquetage dans les données ou la quantité de mélange pour choisir C ($\Rightarrow$ faire une validation croisée)

A savoir

Il existe plusieurs déclinaisons des SVMs adaptées à différentes configurations :

- à la régression

Régression

Les étiquettes sont des réels. La marge correspond alors à un tube autour de la fonction cible.

A savoir

Il existe plusieurs déclinaisons des SVMs adaptées à différentes configurations :

- à la régression
- à l'apprentissage actif

Actif

Les étiquettes sont données à la demande - souvent car cela coûte cher de les obtenir.
Il faut alors déterminer comment sélectionner les points à étiqueter.

- on apprend avec un petit échantillon de points étiquetés
- on regarde les points suivants (au hasard ou dans l'ordre d'arrivée)
- on demande l'étiquette d'un point s'il est proche de la frontière courante
- on met à jour le SVM avec ce point étiqueté

A savoir

Il existe plusieurs déclinaisons des SVMs adaptées à différentes configurations :

- à la régression
- à l'apprentissage actif
- au cas multi-classes

Multi-classes

On combine des SVM binaires :

- un-contre-tous
- un-contre-un
- global (souvent très lent)

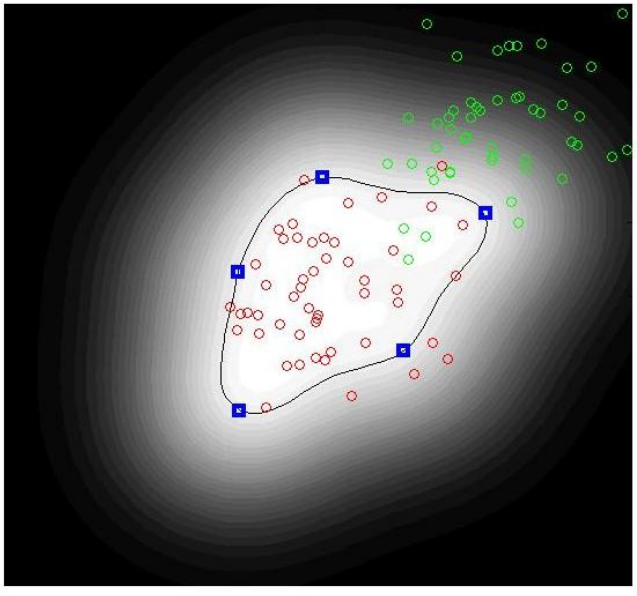
A savoir

Il existe plusieurs déclinaisons des SVMs adaptées à différentes configurations :

- à la régression
- à l'apprentissage actif
- au cas multi-classes
- au cas à une seule classe

Une classe

Sert à la détection d'anomalie ou de nouvelle classe.



Mais...

Il reste encore des progrès à faire !

- Espace mémoire : le calcul et le stockage des éléments du noyau est $\mathcal{O}(n^2)$
- Temps de calcul : le calcul des éléments du noyau est $\mathcal{O}(n^2)$
- Temps de calcul : le calcul des α_i est $\mathcal{O}(k^3)$ si k est le nombre de vecteurs supports

Mais...

Il reste encore des progrès à faire !

- Espace mémoire : le calcul et le stockage des éléments du noyau est $\mathcal{O}(n^2)$
- Temps de calcul : le calcul des éléments du noyau est $\mathcal{O}(n^2)$
- Temps de calcul : le calcul des α_i est $\mathcal{O}(k^3)$ si k est le nombre de vecteurs supports

Réduire la mémoire nécessaire : utiliser un cache ($\Rightarrow$ augmente les temps de calcul).

Mais...

Il reste encore des progrès à faire !

- Espace mémoire : le calcul et le stockage des éléments du noyau est $\mathcal{O}(n^2)$
- Temps de calcul : le calcul des éléments du noyau est $\mathcal{O}(n^2)$
- Temps de calcul : le calcul des α_i est $\mathcal{O}(k^3)$ si k est le nombre de vecteurs supports

Réduire la mémoire nécessaire : utiliser un cache ($\Rightarrow$ augmente les temps de calcul).
Réduire le temps de calcul : recherche une solution approchée !

Plan

SimpleSVM - Contraintes actives

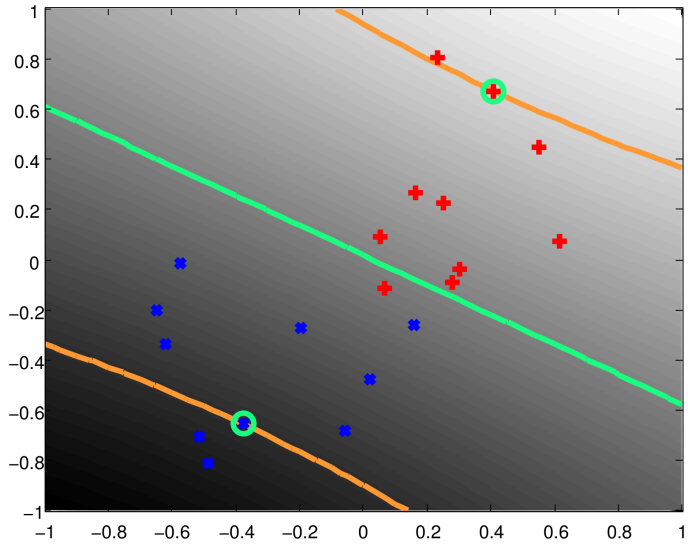
Algorithme

Initialiser
tant que points mal classés **faire**
    calculer α_{I_a}
    si $\alpha_{I_a} \leq 0$ ou $\alpha_{I_a} \geq C$ **alors**
        retirer un point de I_a
    sinon
        si $y(f(x_{I_0}) + b) \leq 1$ ou $y(f(x_{I_C}) + b) \geq 1$
alors
            ajouter un point à I_a
        fin si
    fin si
fin tant que

SimpleSVM - Contraintes actives

Algorithme

Initialiser
tant que points mal classés **faire**
    calculer α_{I_a}
    si $\alpha_{I_a} \leq 0$ ou $\alpha_{I_a} \geq C$ **alors**
        retirer un point de I_a
    sinon
        si $y(f(x_{I_0}) + b) \leq 1$ ou $y(f(x_{I_C}) + b) \geq 1$
alors
            I_C ou I_0 vers I_a
        fin si
    fin si
fin tant que

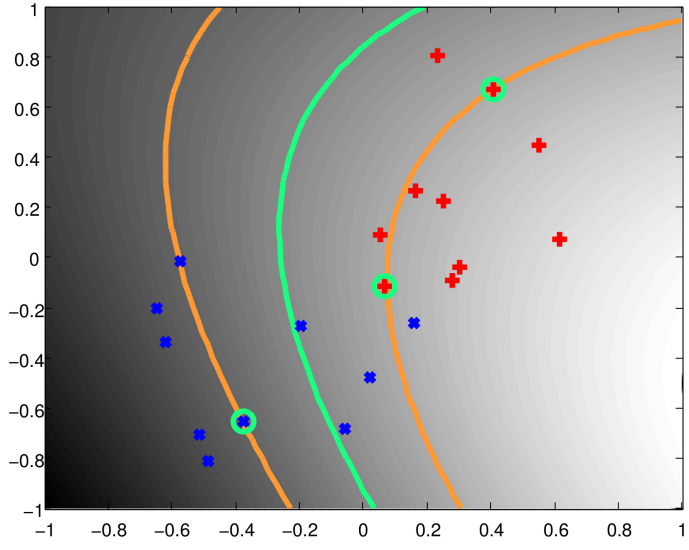


$$\alpha_{I_a} = \{4.7504; 4.7504\}$$

SimpleSVM - Contraintes actives

Algorithme

Initialiser
tant que points mal classés **faire**
  calculer α_{I_a}
  si $\alpha_{I_a} \leq 0$ ou $\alpha_{I_a} \geq C$ **alors**
    I_a vers I_C ou I_0
  sinon
    si $y(f(x_{I_0}) + b) \leq 1$ ou $y(f(x_{I_C}) + b) \geq 1$
alors
      ajouter un point à I_a
    fin si
  fin si
fin tant que

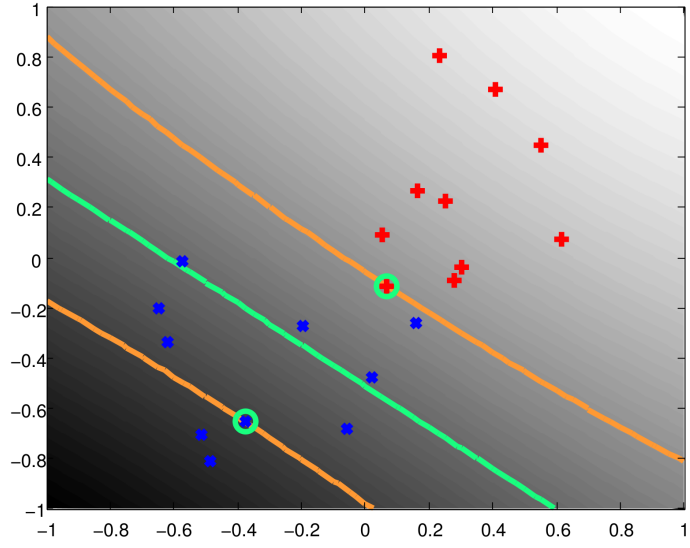


$$\alpha_{I_a} = \{58.2821; -38.3212; 96.6033\}$$

SimpleSVM - Contraintes actives

Algorithme

Initialiser
tant que points mal classés **faire**
  calculer α_{I_a}
  si $\alpha_{I_a} \leq 0$ ou $\alpha_{I_a} \geq C$ **alors**
    retirer un point de I_a
  sinon
    si $y(f(x_{I_0}) + b) \leq 1$ ou $y(f(x_{I_C}) + b) \geq 1$
alors
      I_C ou I_0 vers I_a
    fin si
  fin si
fin tant que

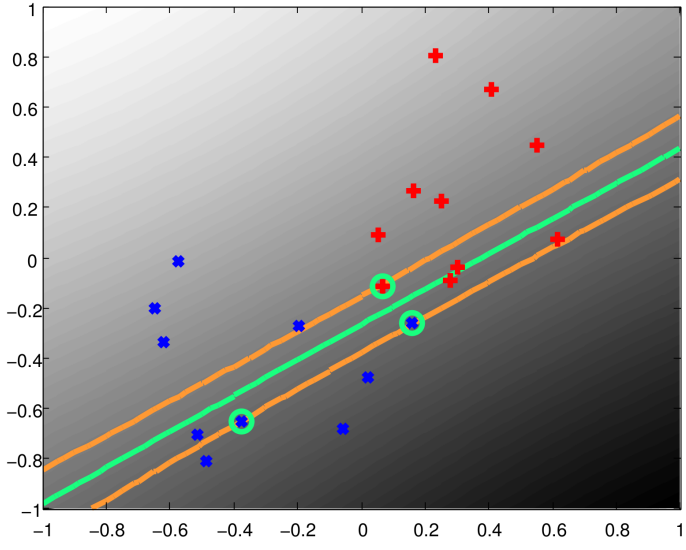


$$\alpha_{I_a} = \{21.0157; 21.0157\}$$

SimpleSVM - Contraintes actives

Algorithme

Initialiser
tant que points mal classés **faire**
    calculer α_{I_a}
    si $\alpha_{I_a} \leq 0$ ou $\alpha_{I_a} \geq C$ **alors**
        I_a vers I_C ou I_0
    sinon
        si $y(f(x_{I_0}) + b) \leq 1$ ou $y(f(x_{I_C}) + b) \geq 1$
alors
            ajouter un point à I_a
        fin si
    fin si
fin tant que

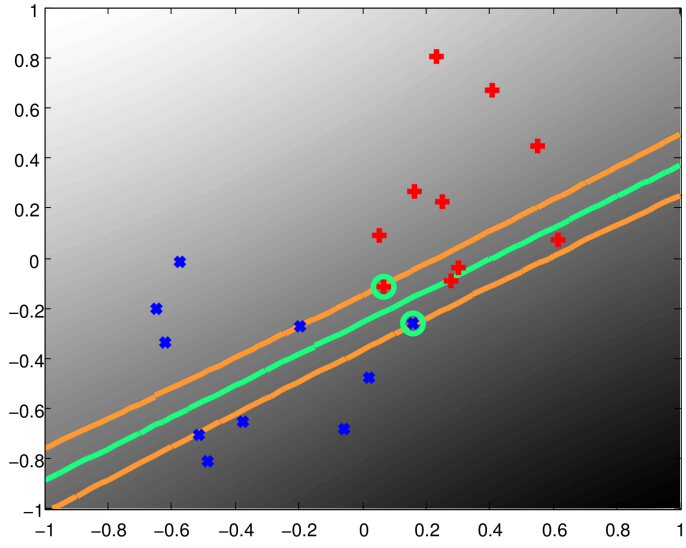


$$\alpha_{I_a} = \{-5.5362; 343.3578; 337.8216\}$$

SimpleSVM - Contraintes actives

Algorithme

Initialiser
tant que points mal classés **faire**
    calculer α_{I_a}
    si $\alpha_{I_a} \leq 0$ ou $\alpha_{I_a} \geq C$ **alors**
        retirer un point de I_a
    sinon
        si $y(f(x_{I_0}) + b) \leq 1$ ou $y(f(x_{I_C}) + b) \geq 1$
alors
            I_C ou I_0 vers I_a
        fin si
    fin si
fin tant que

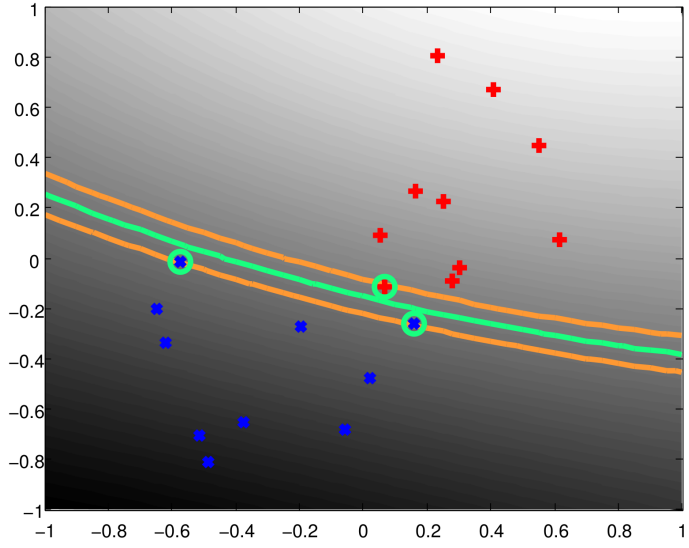


$$\alpha_{I_a} = \{336.5028; 336.5028\}$$

SimpleSVM - Contraintes actives

Algorithme

Initialiser
tant que points mal classés **faire**
    calculer α_{I_a}
    si $\alpha_{I_a} \leq 0$ ou $\alpha_{I_a} \geq C$ **alors**
        retirer un point de I_a
    sinon
        si $y(f(x_{I_0}) + b) \leq 1$ ou $y(f(x_{I_C}) + b) \geq 1$
alors
            ajouter un point à I_a
        fin si
    fin si
fin tant que



$$\alpha_{I_a} = \{641.3429; 773.3408; 131.9979\}$$

SimpleSVM - Contraintes actives

Algorithme

Initialiser
tant que points mal classés **faire**
  calculer α_{I_a}
  si $\alpha_{I_a} \leq 0$ ou $\alpha_{I_a} \geq C$ **alors**
    retirer un point de I_a
  sinon
    si $y(f(x_{I_0}) + b) \leq 1$ ou $y(f(x_{I_C}) + b) \geq 1$
alors
      ajouter un point à I_a
    fin si
  fin si
fin tant que