Méthodes à noyau

Introduction à l'apprentissage automatique – GIF-4101 / GIF-7005

Professeur : Christian Gagné

Semaine 6

6.1 Retour sur les discriminants linéaires

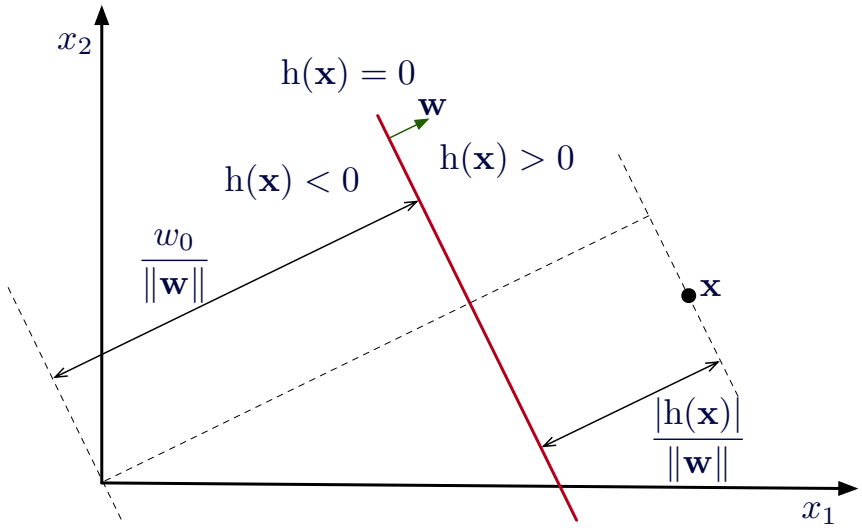
Discriminants linéaires

- Équation d'un discriminant linéaire

$$\mathrm{h}_i(\mathbf{x}|\mathbf{w}_i, w_{i,0}) = \sum_{j=1}^{D} w_{i,j} x_j + w_{i,0}$$

- Modèle à deux classes
  - Une seule équation $\mathrm{h}(\mathbf{x}|\mathbf{w}, w_0)$
  - $\mathbf{x}$ appartient à C_1 si $\mathrm{h}(\mathbf{x}) \geq 0$
  - Autrement (lorsque $\mathrm{h}(\mathbf{x}) < 0$) $\mathbf{x}$ appartient à C_2
  - Poids $\mathbf{w}$ détermine l'orientation de l'hyperplan séparateur
  - Biais w_0 détermine la position de l'hyperplan séparateur dans l'espace d'entrée

Géométrie des discriminants linéaires



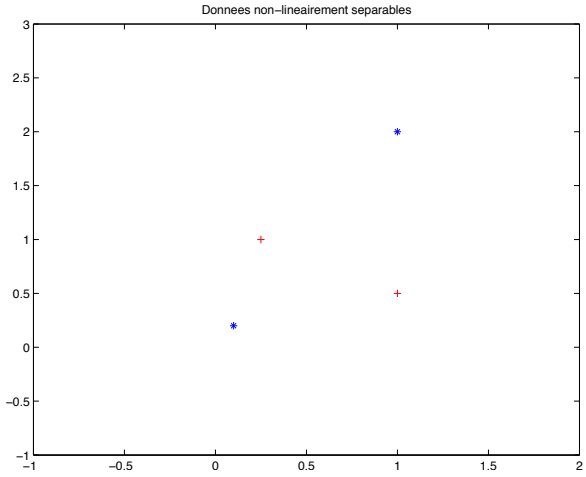
Critère du Perceptron

- Critère du Perceptron

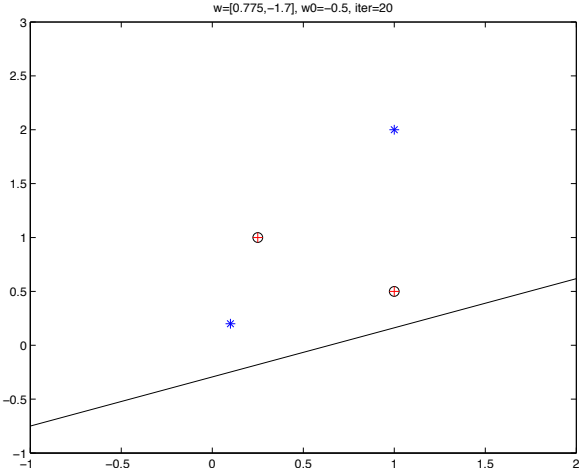
$$
\begin{aligned}
E_{percp}(\mathbf{w}, w_0 | \mathcal{X}) &= -\sum_{\mathbf{x}^t \in \mathcal{Y}} r^t \mathrm{h}(\mathbf{x}^t | \mathbf{w}, w_0) \\
\mathcal{Y} &= \{\mathbf{x}^t \in \mathcal{X} | r^t \mathrm{h}(\mathbf{x}^t | \mathbf{w}, w_0) < 0\}
\end{aligned}
$$

- Faible lien entre l'erreur et la nature des erreurs
  - Classifieur risque de diverger sur données non linéairement séparables

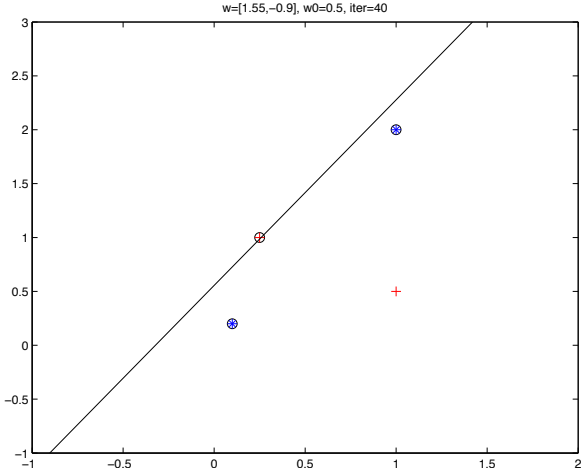
Perceptron sur données non linéairement séparables



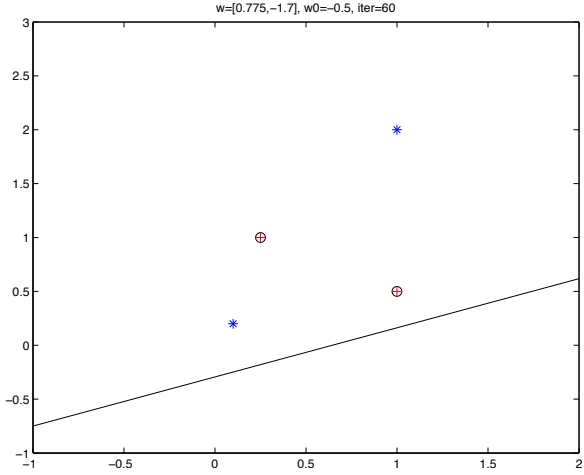
Perceptron sur données non linéairement séparables



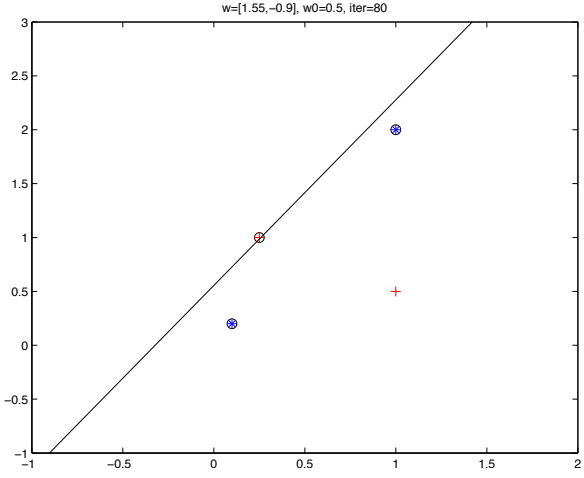
Perceptron sur données non linéairement séparables



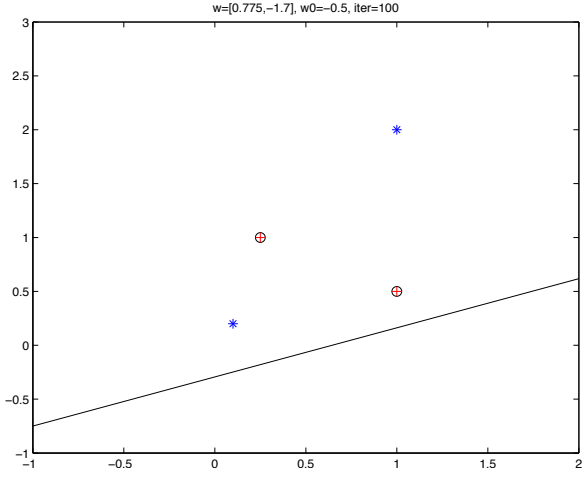
Perceptron sur données non linéairement séparables



Perceptron sur données non linéairement séparables



Perceptron sur données non linéairement séparables



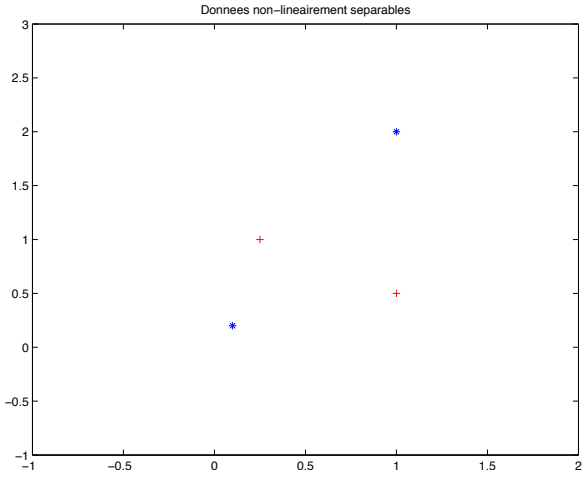
Critère des moindres carrés

- Critère des moindres carrés : régression pour classement

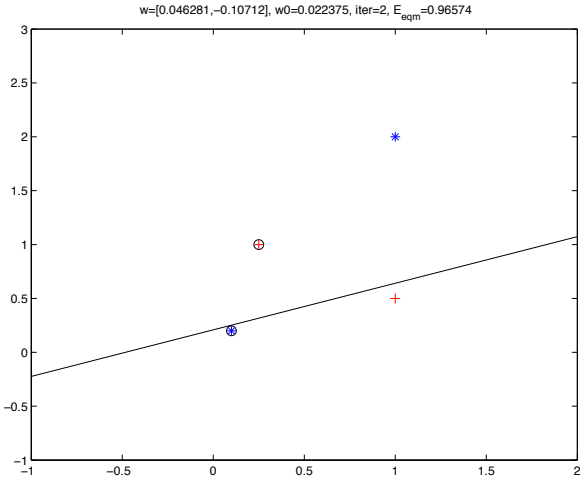
$$E_{quad}(\mathbf{w}, w_0 | \mathcal{X}) = \frac{1}{2} \sum_{\mathbf{x}^t \in \mathcal{X}} (r^t - (\mathbf{w}^\top \mathbf{x}^t + w_0))^2$$

- Tend à minimiser la distance des $\mathrm{h}(\mathbf{x})$ à la valeur r^t
  - Gère mieux les données non linéairement séparables
  - Met l'accent sur les données éloignées de l'hyperplan séparateur

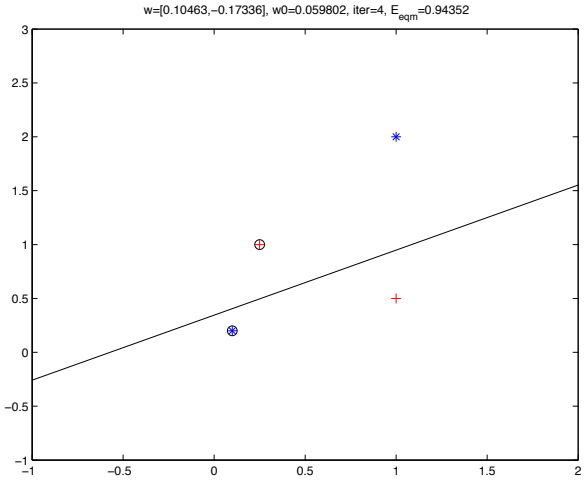
Moindres carrés sur données non linéairement séparables



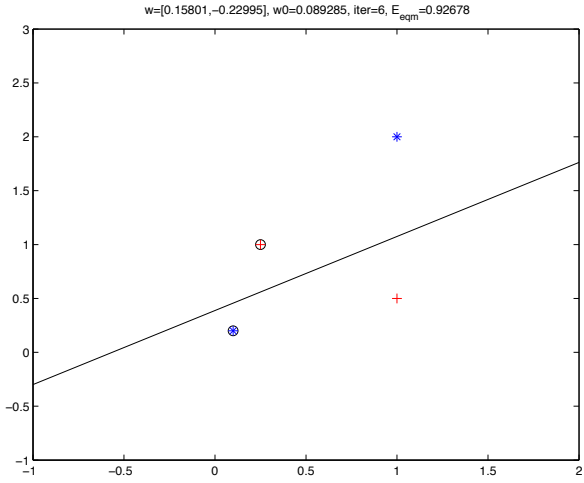
Moindres carrés sur données non linéairement séparables



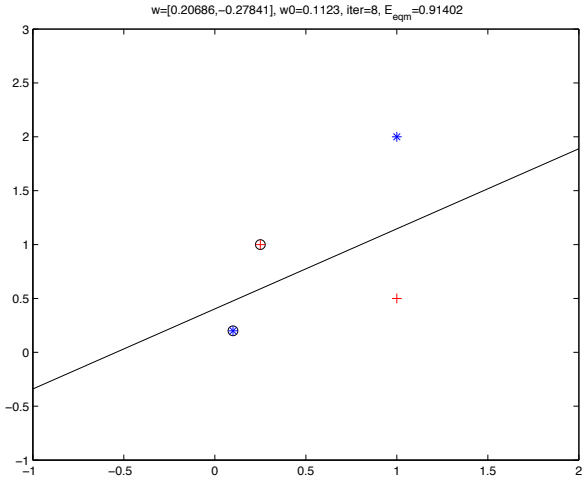
Moindres carrés sur données non linéairement séparables



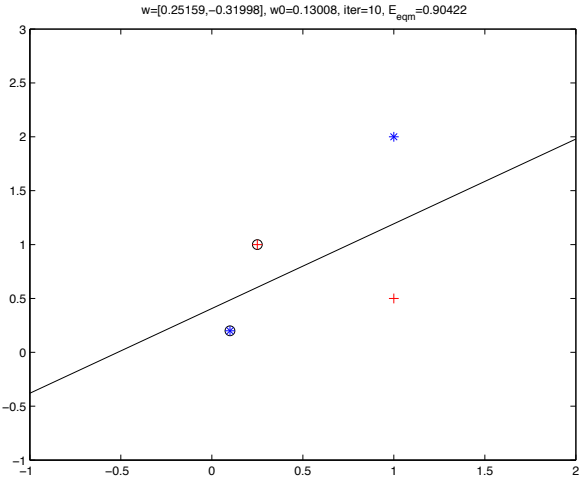
Moindres carrés sur données non linéairement séparables



Moindres carrés sur données non linéairement séparables



Moindres carrés sur données non linéairement séparables

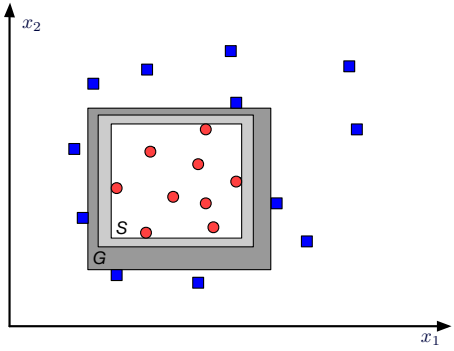


6.2 Séparateurs à vastes marges

Séparateurs à vastes marges (SVM)

- SVM : séparateurs à vastes marges
  - En anglais : *support vector machines*
- Maximisation des marges géométriques
  - Vise un placement optimal de l'hyperplan séparateur
  - Arguments en théorie de l'apprentissage computationnel que ce critère minimise l'erreur (selon l'espace des versions)
- Développement pour un discriminant linéaire
  - Extension à des modèles non linéaires par des fonctions noyau

Espace des versions



- G : hypothèse la plus générale
- S : hypothèse la plus spécifique
- Hypothèses dans $\mathcal{H}$ entre S et G font parties de l'*espace des versions*

Maximisation des marges géométriques

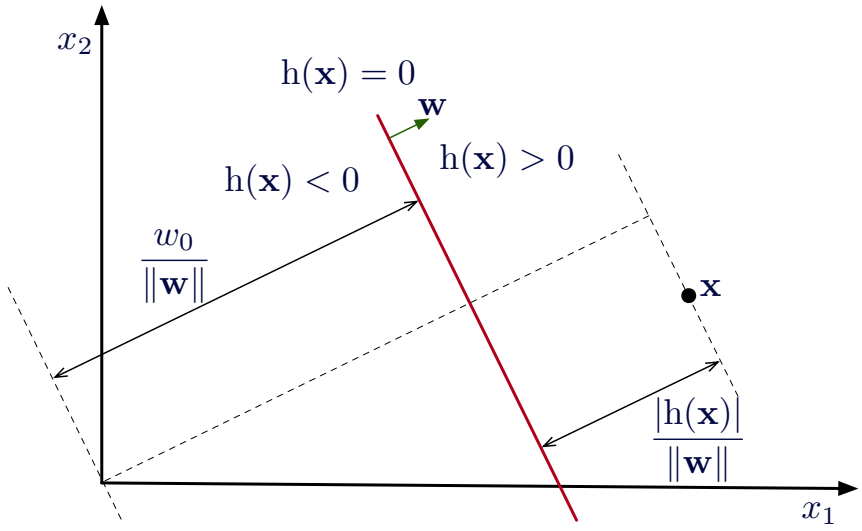
- Recherche de poids $\mathbf{w}$ et w_0 maximisant la marge géométrique pour un jeu de données $\mathcal{X} = \{\mathbf{x}^t, r^t\}$, où $r^t \in \{-1, +1\}$
- Distances à l'hyperplan séparateur des données

$$\frac{|\mathbf{w}^\top \mathbf{x}^t + w_0|}{\|\mathbf{w}\|} = \frac{r^t(\mathbf{w}^\top \mathbf{x}^t + w_0)}{\|\mathbf{w}\|}$$

- On veut cette distance supérieure à un seuil ρ (marge) pour toutes les données

$$\frac{r^t(\mathbf{w}^\top \mathbf{x}^t + w_0)}{\|\mathbf{w}\|} \geq \rho, \forall t$$

Géométrie des discriminants linéaires



Maximisation des marges géométriques

- $\mathbf{w}^\top \mathbf{x}^t + w_0$ est sous-déterminée, il y a une infinité de solutions

$$\underset{w_0 = 1}{\mathbf{w}^\top = \begin{bmatrix} 2 \\ 0{,}5 \end{bmatrix}} \equiv \underset{w_0 = 0{,}5}{\mathbf{w}^\top = \begin{bmatrix} 1 \\ 0{,}25 \end{bmatrix}} \equiv \underset{w_0 = 10}{\mathbf{w}^\top = \begin{bmatrix} 20 \\ 5 \end{bmatrix}}$$

- On pose que $\rho\|\mathbf{w}\| = 1$, ce qui donne :

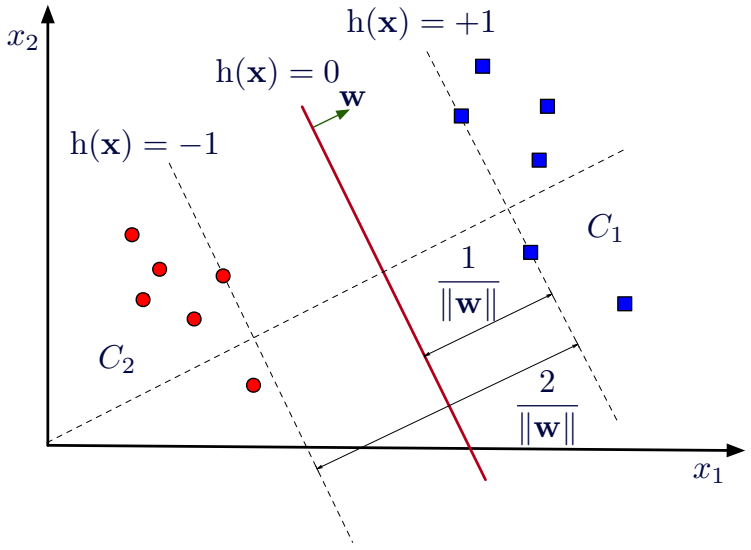
$$\begin{aligned} \mathbf{w}^\top \mathbf{x}^t + w_0 &\geq +1 \quad \text{pour} \quad r^t = +1 \\ \mathbf{w}^\top \mathbf{x}^t + w_0 &\leq -1 \quad \text{pour} \quad r^t = -1 \end{aligned}$$

- Formulation équivalente

$$r^t(\mathbf{w}^\top \mathbf{x}^t + w_0) \geq +1$$

- Minimisation de $\|\mathbf{w}\|$ permet une maximisation de la marge ρ

Maximisation des marges géométriques



6.3 Problème d'optimisation des SVM

Multiplicateurs de Lagrange

- Méthode de résolution de problèmes d'optimisation sous contraintes
  - Exemple : maximiser $f(\mathbf{x})$ sous contraintes que $g(\mathbf{x}) = 0$
  - Il existe un paramètre $\lambda \neq 0$ de sorte que

$$\nabla f + \lambda \nabla g = 0$$

  - Équation correspondante avec multiplicateur de Lagrange

$$L(\mathbf{x},\lambda) \equiv f(\mathbf{x}) + \lambda g(\mathbf{x})$$

- Maximum obtenu en trouvant $\nabla L(\mathbf{x},\lambda) = 0$
  - Si on est intéressé uniquement au $\mathbf{x}$, on peut éliminer λ sans devoir l'évaluer

Exemple avec le multiplicateur de Lagrange

- Maximiser $f(x_1,x_2) = 1 - x_1^2 - x_2^2$ sujet à la contrainte $g(x_1,x_2) = x_1 + x_2 - 1 = 0$
- Formulation avec multiplicateur de Lagrange

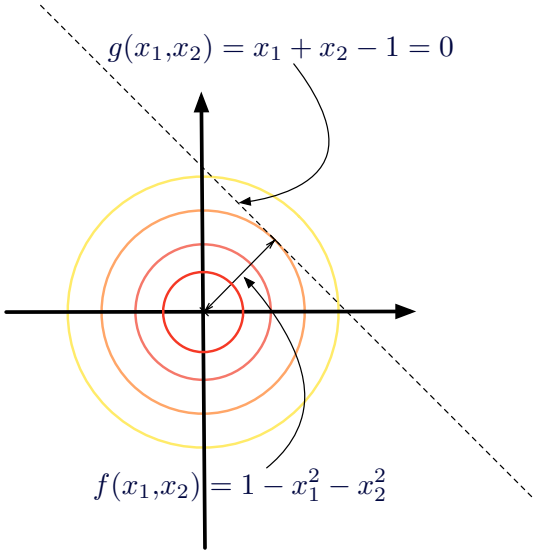
$$L(x_1,x_2,\lambda) = 1 - x_1^2 - x_2^2 + \lambda(x_1 + x_2 - 1)$$

- Résolution de $\nabla L(x_1,x_2,\lambda) = 0$

$$\begin{aligned} \frac{\partial L}{\partial x_1} &= -2x_1 + \lambda = 0 \\ \frac{\partial L}{\partial x_2} &= -2x_2 + \lambda = 0 \\ \frac{\partial L}{\partial \lambda} &= x_1 + x_2 - 1 = 0 \end{aligned}$$

- Solution au système d'équations : $x_1 = 0{,}5$, $x_2 = 0{,}5$ et $\lambda = 1$

Exemple avec le multiplicateur de Lagrange



Multiplicateurs de Lagrange avec inégalités

- Contraintes comme inégalités $g(\mathbf{x}) \geq 0$
  - Possibilité 1 : contrainte inactive, $f(\mathbf{x})$ est maximum pour $g(\mathbf{x}) > 0$, donc maximum à $\nabla f(\mathbf{x}) = 0$, ce qui implique $\lambda = 0$
  - Possibilité 2 : contrainte active, $f(\mathbf{x})$ est maximum pour $g(\mathbf{x}) = 0$
    - Dans ce cas, $\nabla f(\mathbf{x}) = -\lambda \nabla g(\mathbf{x})$ et $\lambda > 0$
- Conditions correspondantes (Karush-Kuhn-Tucker)

$$\begin{aligned} g(\mathbf{x}) &\geq 0 \\ \lambda &\geq 0 \\ \lambda g(\mathbf{x}) &= 0 \end{aligned}$$

- Formulation où on minimise $f(\mathbf{x})$, sujet à $g(\mathbf{x}) \geq 0$ (soustraction de la contrainte)

$$L(\mathbf{x},\lambda) = f(\mathbf{x}) - \lambda g(\mathbf{x}), \text{ avec } \lambda \geq 0$$

Formulation du problème d'optimisation du SVM

- Problème d'optimisation du SVM

$$\begin{aligned} \text{minimiser} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{sujet à} \quad & r^t(\mathbf{w}^\top \mathbf{x}^t + w_0) \geq +1, \forall t \end{aligned}$$

- Forme standard d'un problème de programmation quadratique
  - Méthodes (et résolveurs) disponibles pour une résolution exacte du problème
- Reformulation du problème avec utilisation de multiplicateurs de Lagrange (α^t)

$$\begin{aligned} L_p &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_t \alpha^t [r^t(\mathbf{w}^\top \mathbf{x}^t + w_0) - 1] \\ &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_t \alpha^t r^t(\mathbf{w}^\top \mathbf{x}^t + w_0) + \sum_t \alpha^t \end{aligned}$$

Formulations primale et duale

- L_p est la formulation primale du problème

$$L_p = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_t \alpha^t r^t(\mathbf{w}^\top \mathbf{x}^t + w_0) + \sum_t \alpha^t$$

- Résolution de L_p demande de minimiser selon $\{\mathbf{w}, w_0\}$ et maximiser selon $\alpha^t \geq 0$
  - Solution au point de selle selon $\{\mathbf{w}, w_0\}$ et α^t
- Simplification par formulation duale du problème
  - Éliminer $\mathbf{w}$ à l'aide des dérivées partielles de L_p selon $\{\mathbf{w}, w_0\}$ nulles

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0, \quad \frac{\partial L_p}{\partial w_0} = 0$$

Passage à la formulation duale

$$
\begin{aligned}
L_p &= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_t \alpha^t r^t (\mathbf{w}^\top \mathbf{x}^t + w_0) + \sum_t \alpha^t \\
\frac{\partial L_p}{\partial \mathbf{w}} &= \mathbf{w} - \sum_t \alpha^t r^t \mathbf{x}^t = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_t \alpha^t r^t \mathbf{x}^t \\
\frac{\partial L_p}{\partial w_0} &= \sum_t \alpha^t r^t = 0 \\
L_d &= \frac{1}{2}(\mathbf{w}^\top \mathbf{w}) - \mathbf{w}^\top \sum_t \alpha^t r^t \mathbf{x}^t - w_0 \sum_t \alpha^t r^t + \sum_t \alpha^t \\
&= -\frac{1}{2}(\mathbf{w}^\top \mathbf{w}) + \sum_t \alpha^t \\
&= -\frac{1}{2} \sum_t \sum_s \alpha^t \alpha^s r^t r^s (\mathbf{x}^t)^\top \mathbf{x}^s + \sum_t \alpha^t
\end{aligned}
$$

Formulation du problème avec multiplicateurs de Lagrange

- Formulation duale avec multiplicateurs de Lagrange

$$\begin{aligned} \text{maximiser} \quad & -\frac{1}{2}\sum_t \sum_s \alpha^t \alpha^s r^t r^s (\mathbf{x}^t)^\top \mathbf{x}^s + \sum_t \alpha^t \\ \text{sujet à} \quad & \sum_t \alpha^t r^t = 0 \quad \text{et} \quad \alpha^t \geq 0, \forall t \end{aligned}$$

- Nouvelle formulation du problème
  - Taille du problème dépends de la taille du jeu de données (N) plutôt que de la dimensionnalité (D)
- Forme toujours résoluble par programmation quadratique
  - Garantie d'obtenir l'optimum global en temps polynomial
  - Complexité en temps $O(N^3)$, complexité en espace $O(N^2)$
- Formulation permet l'utilisation de fonctions noyau (présenté plus loin)

Vecteurs de support

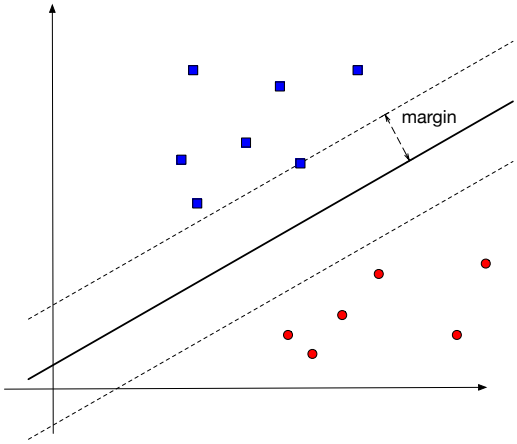
- Nouvelle formulation : un α^t par donnée d'entraînement
  - Normalement, une majorité de $\alpha^t = 0$
  - Les données dont $\alpha^t > 0$ sont les *vecteurs de support*
- Calcul de w_0 à partir des vecteurs de support, $\mathcal{M} = \{\alpha^t | \alpha^t > 0, \forall t\}$

$$w_0 = \mathbb{E}[r^t - \mathbf{w}^\top \mathbf{x}^t] = \frac{1}{|\mathcal{M}|} \sum_{\alpha^t \in \mathcal{M}} \left(r^t - \sum_{\alpha^s \in \mathcal{M}} \alpha^s r^s (\mathbf{x}^t)^\top \mathbf{x}^s \right)$$

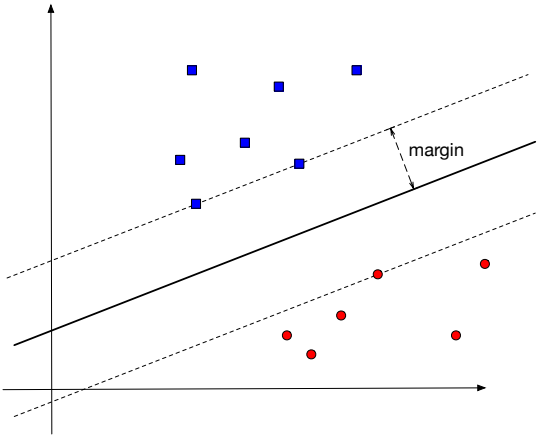
- Évaluation de données après entraînement

$$\mathrm{h}(\mathbf{x}) = \sum_t \alpha^t r^t (\mathbf{x}^t)^\top \mathbf{x} + w_0$$

Illustration des vecteurs de support



Marge sous-optimale



Marge maximimale

6.4 Marges douces

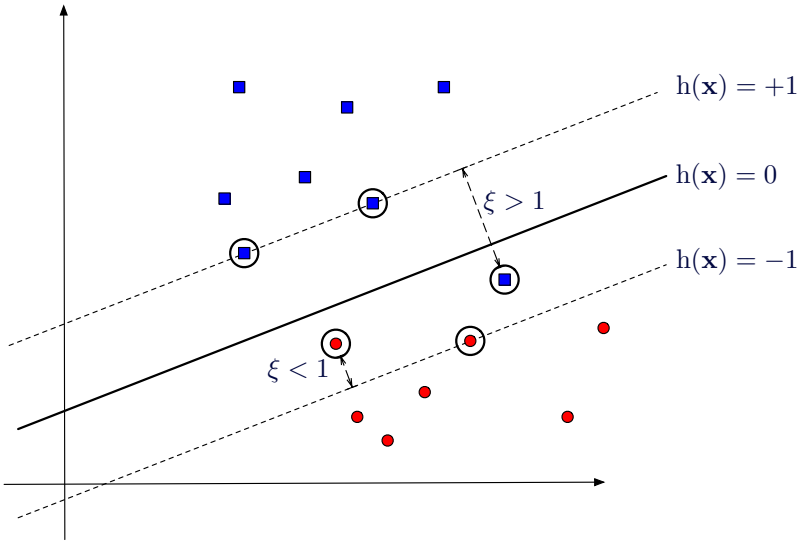
Marges douces

- Dans la formulation actuelle, le SVM reste un discriminant linéaire
  - Avec données non linéairement séparables, aucune solution valide ne peut être obtenue par programmation quadratique
- Introduction de variables *slacks* ($\xi^t \geq 0$) pour chaque donnée $\mathbf{x}^t$
  - Si $\xi^t = 0$, pas de problème avec variable $\mathbf{x}^t$
  - Si $\xi^t > 0$, déviation de la variable $\mathbf{x}^t$ de la marge
    - $0 < \xi^t < 1$: donnée du bon côté, mais dans la marge
    - $\xi^t > 1$: donnée du mauvais côté de l'hyperplan, mal classée
  - Réécriture du critère d'optimisation des SVM

$$r^t(\mathbf{w}^\top \mathbf{x}^t + w_0) \geq 1 - \xi^t$$

- Permet de tolérer des erreurs
  - Erreur associée aux données non séparables : $\sum_t \xi^t$

Marges douces



Reformulation avec marges douces

- Formulation primale avec marges douces

$$L_p = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_t \xi^t - \sum_t \alpha^t[r^t(\mathbf{w}^\top \mathbf{x}^t + w_0) - 1 + \xi^t] - \sum_t \mu^t \xi^t$$

  - μ^t : multiplicateurs de Lagrange pour contraintes $\xi^t \geq 0$
  - C : facteur de pénalité pour régularisation selon les erreurs ξ^t

- Formulation duale avec marges douces

$$\begin{aligned} \text{maximiser} \quad & -\frac{1}{2}\sum_t \sum_s \alpha^t \alpha^s r^t r^s (\mathbf{x}^t)^\top \mathbf{x}^s + \sum_t \alpha^t \\ \text{sujet à} \quad & \sum_t \alpha^t r^t = 0 \quad \text{et} \quad 0 \leq \alpha^t \leq C, \forall t \end{aligned}$$

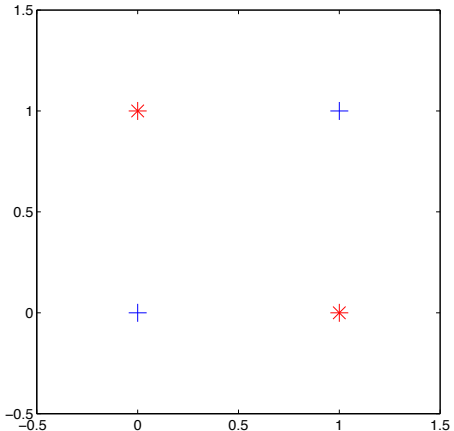
6.5 Retour sur les fonctions de base

Problème du XOR

- Problème du XOR

$$\mathbf{x}_1 = [0\ 0]^\top \quad r_1 = 0$$
$$\mathbf{x}_2 = [0\ 1]^\top \quad r_2 = 1$$
$$\mathbf{x}_3 = [1\ 0]^\top \quad r_3 = 1$$
$$\mathbf{x}_4 = [1\ 1]^\top \quad r_4 = 0$$

- Exemple de données non linéairement séparables



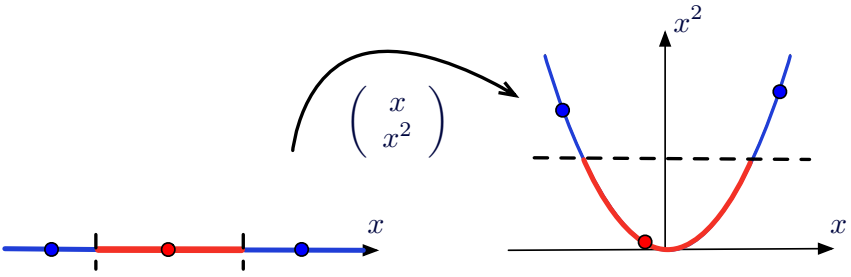
Fonctions de base

- Discriminant avec fonction de base
  - Transformation non linéaire $\phi : \mathbb{R}^D \to \mathbb{R}^K$ écrite sous une forme linéaire

$$\mathrm{h}_i(\mathbf{x}) = \sum_{j=1}^{K} w_j \phi_{i,j}(\mathbf{x}) + w_0$$

  - Exemple de fonctions de base
    - $\phi_{i,j}(\mathbf{x}) = x_j$
    - $\phi_{i,j}(\mathbf{x}) = x_1^{j-1}$
    - $\phi_{i,j}(\mathbf{x}) = \exp(-(x_2 - m_j)^2/c)$
    - $\phi_{i,j}(\mathbf{x}) = \exp(-\|\mathbf{x} - \mathbf{m}_j\|^2/c)$
    - $\phi_{i,j}(\mathbf{x}) = \mathrm{sgn}(x_j - c_j)$

Projection avec une fonction de base



- En 1D : non linéairement séparable
- Avec projection en 2D : linéairement séparable

Fonctions de base

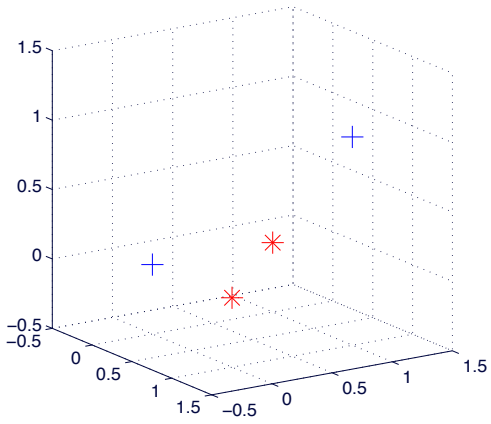
- Résolution du XOR avec fonction de base $\phi : \mathbb{R}^2 \to \mathbb{R}^3$

$$\phi(\mathbf{x}) = [x_1 \; x_2 \; (x_1 x_2)]^\top$$

- Résultats de la transformation

$$\begin{aligned} \mathbf{z}_1 &= [0\ 0\ 0]^\top \quad r_1 = 0 \\ \mathbf{z}_2 &= [0\ 1\ 0]^\top \quad r_2 = 1 \\ \mathbf{z}_3 &= [1\ 0\ 0]^\top \quad r_3 = 1 \\ \mathbf{z}_4 &= [1\ 1\ 1]^\top \quad r_4 = 0 \end{aligned}$$

- Données linéairement séparables dans le nouvel espace !



Fonctions de base radiale

- Fonctions de base radiale (RBF : *Radial Basis Functions*)

$$\phi_i(\mathbf{x}) = \exp\left[-\frac{\|\mathbf{x} - \mathbf{m}_i\|^2}{2s_i^2}\right]$$

- Consiste en une fonction gaussienne centrée sur $\mathbf{m}_i$ avec une influence locale paramétrée par s_i
  - À strictement parler, ce n'est pas une densité de probabilité de loi multinormale ($\int_{-\infty}^{\infty} \phi_i(\mathbf{x}) d\mathbf{x} \neq 1$)
- Idée : chaque fonction gaussienne capture un groupe de données dans un certain voisinage
- Avec K fonctions gaussiennes, projection dans un espace à K dimensions

$$\phi = [\phi_1 \ \ldots \ \phi_K]^\top : \mathbb{R}^D \to \mathbb{R}^K$$

6.6 SVM à noyau

Fonctions de base et SVM

- Transformation non linéaire $\phi : \mathbb{R}^D \to \mathbb{R}^K$ avec fonctions de base

$$\mathbf{z}(\mathbf{x}) = \phi(\mathbf{x})$$

- Discrimination linéaire dans un espace non linéaire

$$\begin{aligned} \mathrm{h}(\mathbf{z}) &= \mathbf{w}^\top \mathbf{z} + w_0 \\ &= \mathbf{w}^\top \phi(\mathbf{x}) + w_0 = \sum_{j=1}^{K} w_j \phi_j(\mathbf{x}) + w_0 \end{aligned}$$

- Reformulation dans la forme duale

$$\begin{aligned} \mathbf{w} &= \sum_t \alpha^t r^t \mathbf{z}^t = \sum_t \alpha^t r^t \phi(\mathbf{x}^t) \\ \mathrm{h}(\mathbf{x}) &= \sum_t \mathbf{w}^\top \phi(\mathbf{x}) + w_0 = \sum_t \alpha^t r^t (\phi(\mathbf{x}^t))^\top \phi(\mathbf{x}) + w_0 \end{aligned}$$

Fonctions noyau

- Fonction noyau : $K(\mathbf{x},\mathbf{y}) = (\phi(\mathbf{x}))^\top \phi(\mathbf{y})$
- SVM avec fonction noyau

$$\mathrm{h}(\mathbf{x}) = \sum_t \alpha^t r^t K(\mathbf{x}^t,\mathbf{x}) + w_0$$

- Truc du noyau : aucun calcul directement dans l'espace généré par $\phi(\mathbf{x})$
  - Permet de traiter des fonctions noyau générant des espaces à haute dimensionnalité (possiblement infinie), sans travailler directement dans ces espaces
- Noyaux couramment utilisés
  - Produit scalaire : $K(\mathbf{x},\mathbf{y}) =< \mathbf{x},\mathbf{y} >= \mathbf{x}^\top \mathbf{y}$
  - Polynomial d'ordre q : $K(\mathbf{x},\mathbf{y}) = (\mathbf{x}^\top \mathbf{y} + 1)^q$
  - Gaussien : $K(\mathbf{x},\mathbf{y}) = \exp\left[-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma^2}\right]$
  - Sigmoïdal : $K(\mathbf{x},\mathbf{y}) = \tanh(2\mathbf{x}^\top \mathbf{y} + 1)$

SVM à noyau

- Entraînement sur jeu de données $\mathcal{X} = \{\mathbf{x}^t, r^t\}_{t=1}^N$
  - Calcul des α^t par programmation quadratique

$$\text{maximiser} \quad L_d = -\frac{1}{2}\sum_{t=1}^{N}\sum_{s=1}^{N}\alpha^t\alpha^s r^t r^s K(\mathbf{x}^t, \mathbf{x}^s) + \sum_t \alpha^t$$

$$\text{sujet à} \quad \sum_t \alpha^t r^t = 0 \quad \text{et} \quad 0 \leq \alpha^t \leq C, \forall t$$

  - Calcul du biais w_0 avec vecteurs de support, $\mathcal{M} = \{\alpha^t | \alpha^t \geq 0, \forall t\}$

$$w_0 = \frac{1}{|\mathcal{M}|}\sum_{\alpha^t \in \mathcal{M}}\left(r^t - \sum_{\alpha^s \in \mathcal{M}} \alpha^s r^s K(\mathbf{x}^t, \mathbf{x}^s)\right)$$

- Évaluation d'une donnée $\mathbf{x}$

$$\mathrm{h}(\mathbf{x}) = \sum_t \alpha^t r^t K(\mathbf{x}^t, \mathbf{x}) + w_0$$

Noyau polynomial

- Noyau polynomial d'ordre q

$$K(\mathbf{x},\mathbf{y}) = (\mathbf{x}^\top \mathbf{y} + 1)^q$$

- Exemple en dimension $D = 2$ et ordre $q = 2$

$$\begin{aligned} K(\mathbf{x},\mathbf{y}) &= (\mathbf{x}^\top \mathbf{y} + 1)^2 = (x_1 y_1 + x_2 y_2 + 1)^2 \\ &= 1 + 2x_1 y_1 + 2x_2 y_2 + 2x_1 x_2 y_1 y_2 + x_1^2 y_1^2 + x_2^2 y_2^2 \end{aligned}$$

  - Fonctions de base correspondantes

$$\begin{aligned} \phi(\mathbf{x}) &= [1 \;\; \sqrt{2}x_1 \;\; \sqrt{2}x_2 \;\; \sqrt{2}x_1 x_2 \;\; x_1^2 \;\; x_2^2]^\top \\ K(\mathbf{x},\mathbf{y}) &= \phi(\mathbf{x})^\top \phi(\mathbf{y}) = (\mathbf{x}^\top \mathbf{y} + 1)^2 \end{aligned}$$

Noyau gaussien

- Noyau gaussien avec étalement σ

$$K(\mathbf{x},\mathbf{y}) = \exp\left[-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma^2}\right]$$

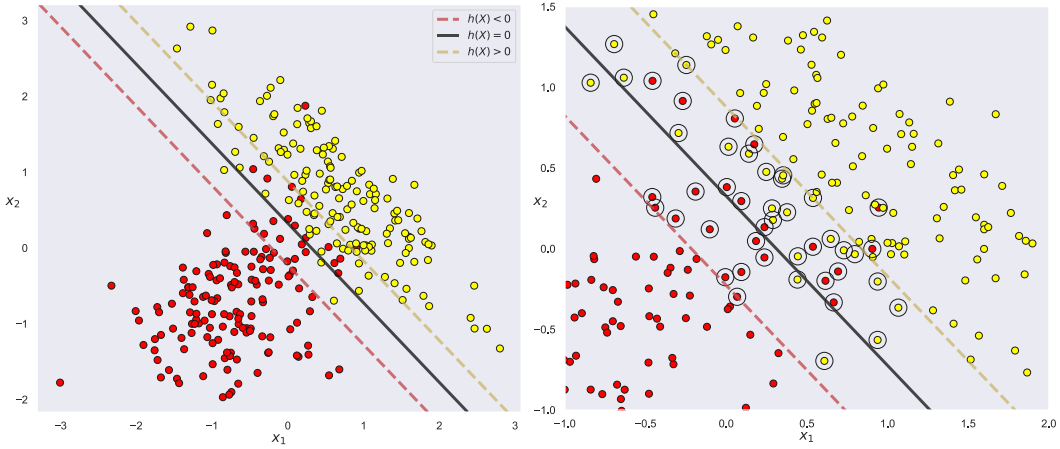
- SVM avec noyau gaussien est un réseau de fonctions RBF entraîné d'une façon particulière

$$\mathrm{h}(\mathbf{x}) = \sum_{t=1}^{N} \alpha^t r^t K(\mathbf{x}^t,\mathbf{x}) + w_0 = \sum_{t=1}^{N} w_t \exp\left[-\frac{\|\mathbf{x}-\mathbf{x}^t\|^2}{\sigma^2}\right] + w_0$$

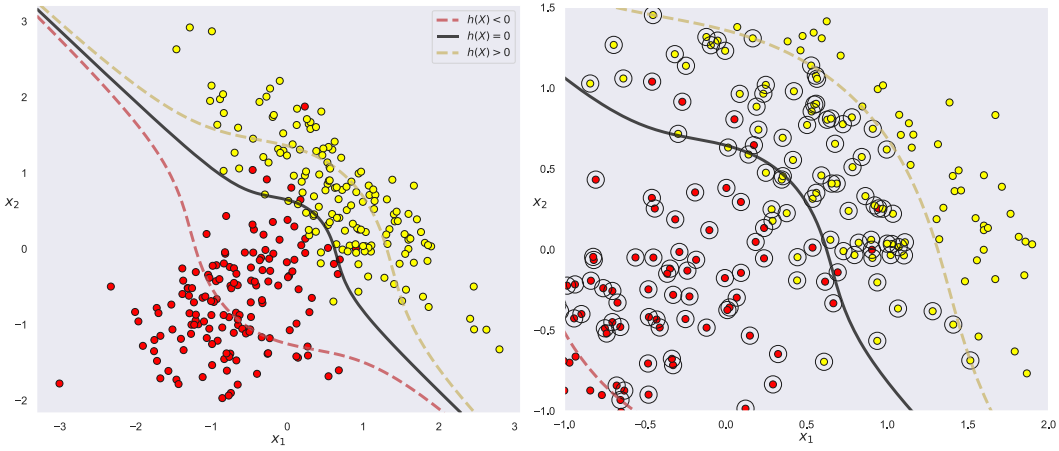
- Estimation de densité par noyau : SVM avec noyau et $\alpha^t = 1, \forall t$

$$\mathrm{h}(\mathbf{x}) = \sum_{t=1}^{N} r^t K(\mathbf{x}^t,\mathbf{x})$$

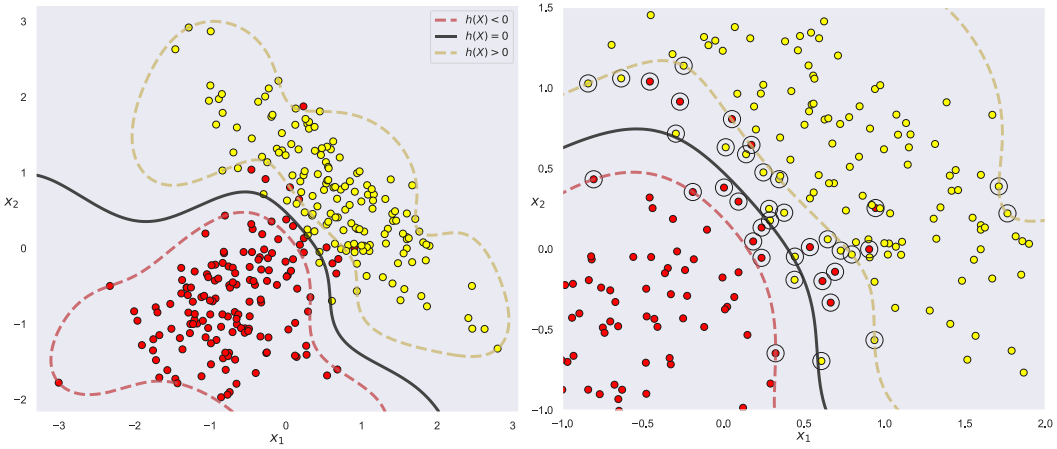
Données avec recoupement : SVM linéaire



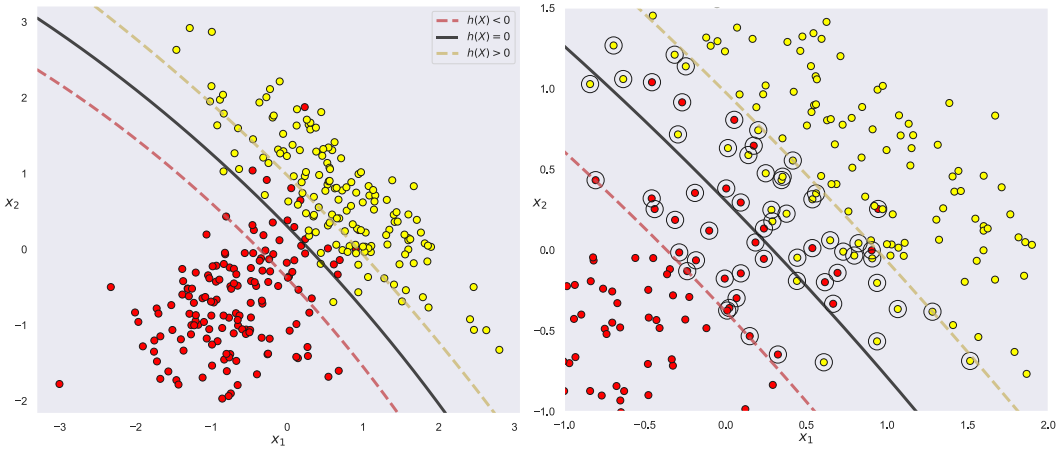
Données avec recoupement : noyau polynomial



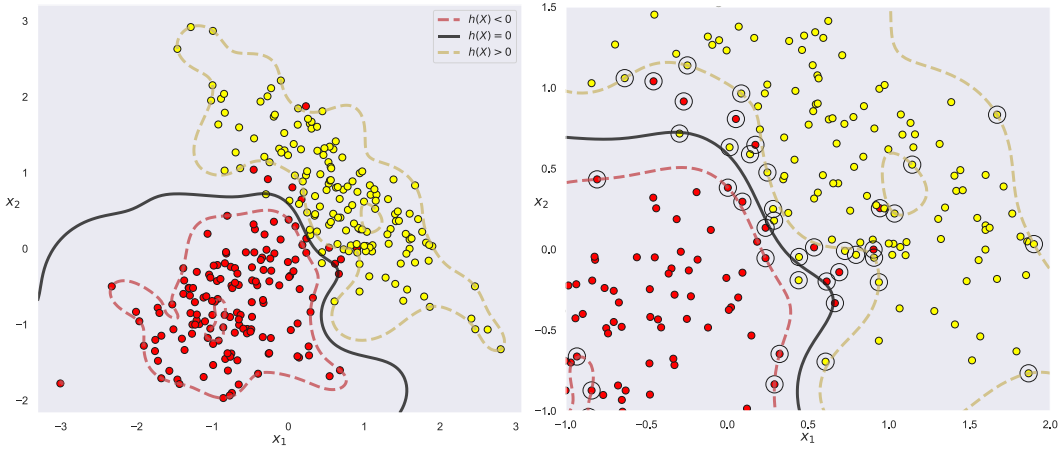
Données avec recoupement : noyau gaussien



Données avec recoupement : noyau gaussien avec grand σ



Données avec recoupement : noyau gaussien avec petit σ



6.7 Hyperparamètres des SVM

Paramètres des SVM

- SVM est une machinerie complexe, où le choix des paramètres peut influencer grandement les résultats
  - Avec noyau gaussien, paramètres C (régularisation) et σ (portée du noyau) ont un impact significatif sur les performances
  - Pour différentes valeurs de ces paramètres, résultats peuvent varier grandement (et parfois être catastrophiques)
  - Ajustement empirique nécessaire, problème par problème
- Règles du pouce pour entraînement de SVM avec noyau gaussien
  - Valeurs de paramètres C à tester : $\{10^{-5}, 10^{-4}, \ldots, 10^{5}\}$
  - Valeurs de paramètres σ à tester : $\{\sigma_{\min}, 2\sigma_{\min}, 4\sigma_{\min}, \ldots, 64\sigma_{\min}\}$ où $\sigma_{\min}$ est la distance euclidienne minimale mesurée entre deux données du jeu de données (excluant les distances nulles) : $\sigma_{\min} = \min_{\forall \mathbf{x}^i \neq \mathbf{x}^j} \|\mathbf{x}^i - \mathbf{x}^j\|$
- Ajustement de ces paramètres par une recherche en grille

Recherche en grille

- Recherche en grille : ajustement de paires de paramètres, avec mesure sur ensemble de validation
  1. Partitionner ensemble de données $\mathcal{X}$ en deux sous-ensembles, $\mathcal{X}_T$ et $\mathcal{X}_V$ (généralement 50%-50%)
  2. Entraîner sommairement classifieur avec $\mathcal{X}_T$ pour chaque paire de paramètres considérés
  3. Sélectionner la paire de paramètres où l'erreur est minimale sur $\mathcal{X}_V$
  4. Utiliser cette paire de paramètres pour entraînement complet sur tout l'ensemble $\mathcal{X}$
- Méthode classique à suivre pour déterminer C et σ des SVM avec noyau gaussien
  - Applicable pour toutes paires de paramètres dont l'effet conjoint est important dans l'entraînement de classifieurs

6.8 SVM par descente du gradient

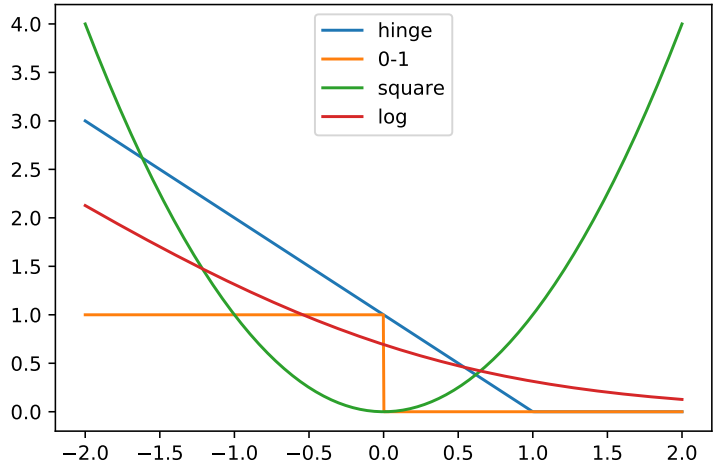
Fonction d'erreur

- SVM : discriminant linéaire avec fonction de perte Hinge

$$\mathcal{L}_{hinge}(y^t, r^t) = \max(1 - y^t r^t, 0)$$

  - $y^t = \mathrm{h}(\mathbf{x}^t | \mathbf{w}, w_0)$
  - Pénalise des données du bon côté de l'hyperplan, mais dans la marge ($y^t r^t < 1$)
- Chaque critère d'erreur fait un compromis différent sur la nature des erreurs
  - Fonction de perte 0/1
  - Erreur quadratique
  - Entropie croisée

Comparaison de différents critères d'erreurs



Descente du gradient avec noyau

- Discriminant dans l'espace généré par un noyau

$$\mathrm{h}(\mathbf{x}) = \sum_{\mathbf{x}^s \in \mathcal{X}} \alpha^s r^s K(\mathbf{x}^s, \mathbf{x}) + w_0$$

- Apprentissage des paramètres α^t et w_0 peut se faire à l'aide d'une descente du gradient
  - Corrections à appliquer aux paramètres du classifieur

$$\Delta\alpha^t = -\eta \frac{\partial E(\boldsymbol{\alpha}, w_0 | \mathcal{X})}{\partial \alpha^t}, \quad \Delta w_0 = -\eta \frac{\partial E(\boldsymbol{\alpha}, w_0 | \mathcal{X})}{\partial w_0}$$

  - Mise à jour, avec contrainte $\alpha^t \geq 0$, $\forall \alpha^t$:

$$\begin{aligned} \alpha^t &= \begin{cases} 0 & \text{si } \alpha^t + \Delta\alpha^t < 0 \\ \alpha^t + \Delta\alpha^t & \text{autrement} \end{cases}, \\ w_0 &= w_0 + \Delta w_0. \end{aligned}$$

Fonction d'erreur pour descente du gradient

- Fonction de perte Hinge avec régularisation pour discriminant avec noyau

$$
\begin{aligned}
E_{hinge}(\boldsymbol{\alpha}, w_0 | \mathcal{X}) &= \sum_{\mathbf{x}^t \in \mathcal{Y}} (1 - r^t \mathrm{h}(\mathbf{x}^t | \boldsymbol{\alpha}, w_0)) + \lambda \frac{1}{2} \sum_{\alpha^s \in \boldsymbol{\alpha}} (\alpha^s)^2 , \\
\mathcal{Y} &= \{\mathbf{x}^t \in \mathcal{X} \mid r^t \mathrm{h}(\mathbf{x}^t | \boldsymbol{\alpha}, w_0) < 1\}.
\end{aligned}
$$

- Effectue une maximisation des marges géométrique dans l'espace des noyaux
  - Valeur $r^t \mathrm{h}(\mathbf{x}^t \mid \boldsymbol{\alpha}, w_0) \in [0,1]$: donnée bien classée, mais dans la marge
- Régularisation est nécessaire
  - Sinon, valeurs des α^t explosent !
  - Paramètre de régularisation λ doit être ajusté empiriquement pour chaque jeu de données (recherche en grille avec le σ pour noyau gaussien)

6.9 Fonctions noyau et distances

Fonctions noyau et distances

- Fonction noyau : mesure de similarité
- Mesure de distance : mesure de dissimilarité
- Distance euclidienne dans l'espace généré par le noyau (espace $\phi(\mathbf{x})$)

$$d(\mathbf{x},\mathbf{y})^2 = K(\mathbf{x},\mathbf{x}) + K(\mathbf{y},\mathbf{y}) - 2K(\mathbf{x},\mathbf{y})$$

- Exemple avec noyau de type produit scalaire, $K(\mathbf{x},\mathbf{y}) = \mathbf{x}^\top\mathbf{y}$

$$\begin{aligned} d(\mathbf{x},\mathbf{y})^2 &= \|\mathbf{x}-\mathbf{y}\|^2 = (\mathbf{x}-\mathbf{y})^\top(\mathbf{x}-\mathbf{y}) \\ &= \mathbf{x}^\top\mathbf{x} + \mathbf{y}^\top\mathbf{y} - 2\mathbf{x}^\top\mathbf{y} \\ &= K(\mathbf{x},\mathbf{x}) + K(\mathbf{y},\mathbf{y}) - 2K(\mathbf{x},\mathbf{y}) \end{aligned}$$

- Permet de faire du classement aux k-plus proches voisins avec des fonctions noyau !
  - Vecteurs de support = sélection de prototypes

Matrice de Gram

- Matrice de Gram $G(\mathcal{X})$: mesure de similarités entre toutes les données de $\mathcal{X} = \{\mathbf{x}^t\}_{t=1}^N$

$$G(\mathcal{X}) = \begin{bmatrix} K(\mathbf{x}^1,\mathbf{x}^1) & K(\mathbf{x}^1,\mathbf{x}^2) & \cdots & K(\mathbf{x}^1,\mathbf{x}^N) \\ K(\mathbf{x}^2,\mathbf{x}^1) & K(\mathbf{x}^2,\mathbf{x}^2) & \cdots & K(\mathbf{x}^2,\mathbf{x}^N) \\ \cdots & \cdots & \ddots & \cdots \\ K(\mathbf{x}^N,\mathbf{x}^1) & K(\mathbf{x}^N,\mathbf{x}^2) & \cdots & K(\mathbf{x}^N,\mathbf{x}^N) \end{bmatrix}$$

- Matrice symétrique
- Forme similaire à une matrice de distances ou une matrice de covariance

6.10 SVM dans scikit-learn

Scikit-learn

- `svm.SVC` : SVM à noyau tel que vu en classe
  - Quelques noyaux standards supportés (linéaire, gaussien, polynomial, sigmoïde), matrice de Gram peut également être fournie
  - Passage à l'échelle plus difficile, ne fonctionne pas bien avec $N > 100\,000$
- `svm.NuSVC` : variante de SVM à noyau
  - Régularisation contrôlant directement nombre de vecteurs de support
- `svm.LinearSVC` : SVM linéaire
  - Optimisé pour SVM linéaire, meilleure utilisation des ressources et meilleur passage à l'échelle
- `linear_model.SGDClassifier` : descente du gradient stochastique
  - Peut émuler SVM linéaire avec bonne configuration de fonction de perte et régularisation
  - Efficace dans l'utilisation des ressources, permet un traitement en ligne