Introduction

Objectifs

1. Comprendre les grands principes des méthodes à noyaux

Objectifs

1. Comprendre les grands principes des méthodes à noyaux
2. Entrevoir les aspects algorithmiques des Support Vector Machines

Objectifs

1. Comprendre les grands principes des méthodes à noyaux
2. Entrevoir les aspects algorithmiques des Support Vector Machines
3. Percevoir l'impact de la théorie pour expliquer leurs performances statistiques

Objectifs

1. Comprendre les grands principes des méthodes à noyaux
2. Entrevoir les aspects algorithmiques des Support Vector Machines
3. Percevoir l'impact de la théorie pour expliquer leurs performances statistiques
4. Observer leur mise en oeuvre pour l'analyse des données

Objectifs

1. Comprendre les grands principes des méthodes à noyaux
2. Entrevoir les aspects algorithmiques des Support Vector Machines
3. Percevoir l'impact de la théorie pour expliquer leurs performances statistiques
4. Observer leur mise en oeuvre pour l'analyse des données

En somme...

montrer les bienfaits d'une interaction permanente entre théorie et algorithmique

Contexte

- explosion de la taille et de la diversité des données

Contexte

- explosion de la taille et de la diversité des données
  - WWW

Contexte

- explosion de la taille et de la diversité des données
  - WWW
  - ADN

Contexte

- explosion de la taille et de la diversité des données
  - WWW
  - ADN
  - marchés financiers

Contexte

- explosion de la taille et de la diversité des données
  - WWW
  - ADN
  - marchés financiers
- disponibilité des données

Contexte

- explosion de la taille et de la diversité des données
  - WWW
  - ADN
  - marchés financiers
- disponibilité des données
  - explosion des capacités de stockage et de calculs

Contexte

- explosion de la taille et de la diversité des données
  - WWW
  - ADN
  - marchés financiers
- disponibilité des données
  - explosion des capacités de stockage et de calculs
  - explosion des communications

Contexte

- explosion de la taille et de la diversité des données
  - WWW
  - ADN
  - marchés financiers
- disponibilité des données
  - explosion des capacités de stockage et de calculs
  - explosion des communications
- digitalisation du monde

Contexte

- explosion de la taille et de la diversité des données
  - WWW
  - ADN
  - marchés financiers
- disponibilité des données
  - explosion des capacités de stockage et de calculs
  - explosion des communications
- digitalisation du monde
  - lutte anti-terroriste

Contexte

- explosion de la taille et de la diversité des données
  - WWW
  - ADN
  - marchés financiers
- disponibilité des données
  - explosion des capacités de stockage et de calculs
  - explosion des communications
- digitalisation du monde
  - lutte anti-terroriste
  - GoogleEarth

Contexte

- explosion de la taille et de la diversité des données
  - WWW
  - ADN
  - marchés financiers
- disponibilité des données
  - explosion des capacités de stockage et de calculs
  - explosion des communications
- digitalisation du monde
  - lutte anti-terroriste
  - GoogleEarth
  - projet "Barcode of Life"

Contexte

- explosion de la taille et de la diversité des données
  - WWW
  - ADN
  - marchés financiers
- disponibilité des données
  - explosion des capacités de stockage et de calculs
  - explosion des communications
- digitalisation du monde
  - lutte anti-terroriste
  - GoogleEarth
  - projet "Barcode of Life"
- besoin de normes quantitatives - contrôle des risques

Contexte

- explosion de la taille et de la diversité des données
  - WWW
  - ADN
  - marchés financiers
- disponibilité des données
  - explosion des capacités de stockage et de calculs
  - explosion des communications
- digitalisation du monde
  - lutte anti-terroriste
  - GoogleEarth
  - projet "Barcode of Life"
- besoin de normes quantitatives - contrôle des risques
  - Bâle II

Contexte

- explosion de la taille et de la diversité des données
  - WWW
  - ADN
  - marchés financiers
- disponibilité des données
  - explosion des capacités de stockage et de calculs
  - explosion des communications
- digitalisation du monde
  - lutte anti-terroriste
  - GoogleEarth
  - projet "Barcode of Life"
- besoin de normes quantitatives - contrôle des risques
  - Bâle II
  - sécurité alimentaire

Constats

- Caractéristique des données : grand nombre de variables

Constats

- Caractéristique des données : grand nombre de variables
- Pour être appliquées, les méthodes statistiques classiques (analyse discriminante, régression linéaire) nécessitent une étape critique de prétraitement

Constats

- Caractéristique des données : grand nombre de variables
- Pour être appliquées, les méthodes statistiques classiques (analyse discriminante, régression linéaire) nécessitent une étape critique de prétraitement
- Les méthodes d'apprentissage fournissent des boîtes noires qui présentent

Constats

- Caractéristique des données : grand nombre de variables
- Pour être appliquées, les méthodes statistiques classiques (analyse discriminante, régression linéaire) nécessitent une étape critique de prétraitement
- Les méthodes d'apprentissage fournissent des boîtes noires qui présentent
  - de hautes performances

Constats

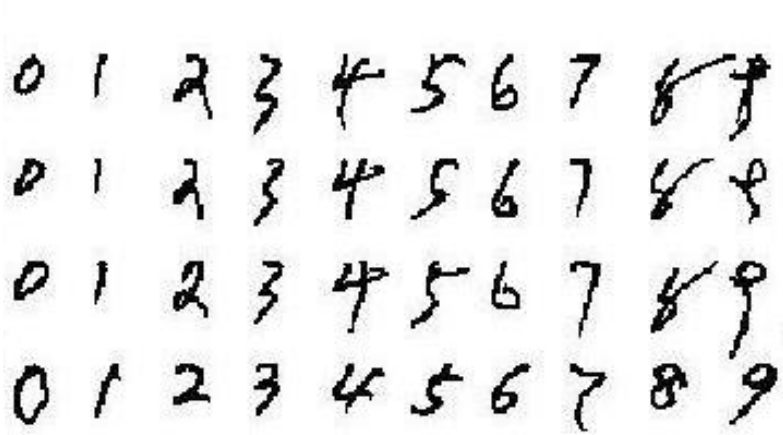
- Caractéristique des données : grand nombre de variables
- Pour être appliquées, les méthodes statistiques classiques (analyse discriminante, régression linéaire) nécessitent une étape critique de prétraitement
- Les méthodes d'apprentissage fournissent des boîtes noires qui présentent
  - de hautes performances
  - pour les données en grande dimension

Constats

- Caractéristique des données : grand nombre de variables
- Pour être appliquées, les méthodes statistiques classiques (analyse discriminante, régression linéaire) nécessitent une étape critique de prétraitement
- Les méthodes d'apprentissage fournissent des boîtes noires qui présentent
  - de hautes performances
  - pour les données en grande dimension
  - sans prétraitement

Champs d'application des méthodes à noyaux

- imagerie/reconnaissance de formes
- biologie/domaine médical
- finance/économie
- marketing, ...



Type de problèmes

- apprentissage supervisé: classification, régression

Type de problèmes

- apprentissage supervisé: classification, régression
- apprentissage non supervisé: clustering, détection d'anomalies

Type de problèmes

- apprentissage supervisé: classification, régression
- apprentissage non supervisé: clustering, détection d'anomalies
- réduction de la dimension

Type de problèmes

- apprentissage supervisé: classification, régression
- apprentissage non supervisé: clustering, détection d'anomalies
- réduction de la dimension
- comparaison de données hétérogènes

Type de problèmes

- apprentissage supervisé: classification, régression
- apprentissage non supervisé: clustering, détection d'anomalies
- réduction de la dimension
- comparaison de données hétérogènes
- séparation de signaux

Type de données

- en grande dimension :

Type de données

- en grande dimension :
  - expression de gènes en biologie

Type de données

- en grande dimension :
  - expression de gènes en biologie
  - fouille de données dans les fichiers logs en marketing

Type de données

- en grande dimension :
  - expression de gènes en biologie
  - fouille de données dans les fichiers logs en marketing
  - indexation de textes ou d'images dans les moteurs de recherche

Type de données

- en grande dimension :
  - expression de gènes en biologie
  - fouille de données dans les fichiers logs en marketing
  - indexation de textes ou d'images dans les moteurs de recherche
- structurées

Type de données

- en grande dimension :
  - expression de gènes en biologie
  - fouille de données dans les fichiers logs en marketing
  - indexation de textes ou d'images dans les moteurs de recherche
- structurées
  - séquences d'ADN

Type de données

- en grande dimension :
  - expression de gènes en biologie
  - fouille de données dans les fichiers logs en marketing
  - indexation de textes ou d'images dans les moteurs de recherche
- structurées
  - séquences d'ADN
  - graphes d'interaction

Type de données

- en grande dimension :
  - expression de gènes en biologie
  - fouille de données dans les fichiers logs en marketing
  - indexation de textes ou d'images dans les moteurs de recherche
- structurées
  - séquences d'ADN
  - graphes d'interaction
- hétérogènes

Type de données

- en grande dimension :
  - expression de gènes en biologie
  - fouille de données dans les fichiers logs en marketing
  - indexation de textes ou d'images dans les moteurs de recherche
- structurées
  - séquences d'ADN
  - graphes d'interaction
- hétérogènes
  - vecteurs

Type de données

- en grande dimension :
  - expression de gènes en biologie
  - fouille de données dans les fichiers logs en marketing
  - indexation de textes ou d'images dans les moteurs de recherche
- structurées
  - séquences d'ADN
  - graphes d'interaction
- hétérogènes
  - vecteurs
  - séquences

Type de données

- en grande dimension :
  - expression de gènes en biologie
  - fouille de données dans les fichiers logs en marketing
  - indexation de textes ou d'images dans les moteurs de recherche
- structurées
  - séquences d'ADN
  - graphes d'interaction
- hétérogènes
  - vecteurs
  - séquences
  - courbes

Type de données

- en grande dimension :
  - expression de gènes en biologie
  - fouille de données dans les fichiers logs en marketing
  - indexation de textes ou d'images dans les moteurs de recherche
- structurées
  - séquences d'ADN
  - graphes d'interaction
- hétérogènes
  - vecteurs
  - séquences
  - courbes
  - graphes

Ingrédients mathématiques

- Optimisation quadratique

Ingrédients mathématiques

- Optimisation quadratique
- Théorie des opérateurs à noyau

Ingrédients mathématiques

- Optimisation quadratique
- Théorie des opérateurs à noyau
- Théorie probabiliste de la classification

Ingrédients mathématiques

- Optimisation quadratique
- Théorie des opérateurs à noyau
- Théorie probabiliste de la classification
- Méthodes statistiques pour l'analyse des données

Repères historiques

- Introduction des noyaux : Aronszajn (1950), Parzen (1962)

Repères historiques

- Introduction des noyaux : Aronszajn (1950), Parzen (1962)
- Noyaux et reconnaissance des formes : Aizerman, Braverman, Rozonoer (1964)

Repères historiques

- Introduction des noyaux : Aronszajn (1950), Parzen (1962)
- Noyaux et reconnaissance des formes : Aizerman, Braverman, Rozonoer (1964)
- Perceptron et classifieurs linéaires : Rosenblatt (1958) Vapnik, Chervonenkis (1964)

Repères historiques

- Introduction des noyaux : Aronszajn (1950), Parzen (1962)
- Noyaux et reconnaissance des formes : Aizerman, Braverman, Rozonoer (1964)
- Perceptron et classifieurs linéaires : Rosenblatt (1958) Vapnik, Chervonenkis (1964)
- Support Vector Machines : Boser, Guyon, Vapnik (1992), Cortes, Vapnik (1995), Vapnik (1995), Schölkopf (1997)

Repères historiques

- Introduction des noyaux : Aronszajn (1950), Parzen (1962)
- Noyaux et reconnaissance des formes : Aizerman, Braverman, Rozonoer (1964)
- Perceptron et classifieurs linéaires : Rosenblatt (1958) Vapnik, Chervonenkis (1964)
- Support Vector Machines : Boser, Guyon, Vapnik (1992), Cortes, Vapnik (1995), Vapnik (1995), Schölkopf (1997)
- Performances statistiques : Blanchard, Bousquet, Massart (2004), Steinwart (2005)

Logiciels

- Environ 40 librairies gratuites!!

Logiciels

- Environ 40 librairies gratuites!!
  - code en C : SVM-Light

Logiciels

- Environ 40 librairies gratuites!!
  - code en C : SVM-Light
  - en ligne: GIST

Logiciels

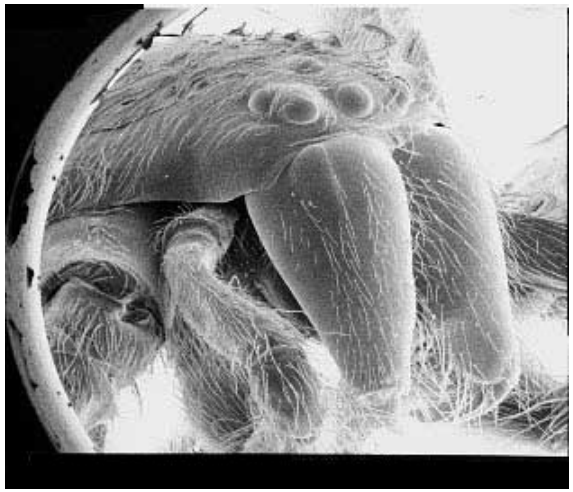
- Environ 40 librairies gratuites!!
  - code en C : SVM-Light
  - en ligne: GIST
  - sous S-Plus : package `libsvm`

Logiciels

- Environ 40 librairies gratuites!!
  - code en C : SVM-Light
  - en ligne: GIST
  - sous S-Plus : package `libsvm`
  - sous R : package `kernlab`

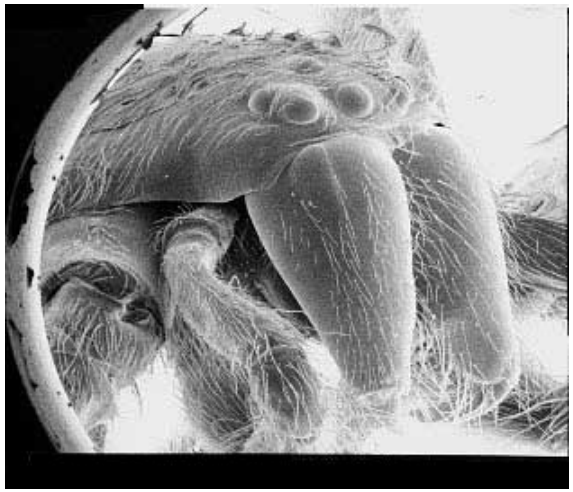
Logiciels

- Environ 40 librairies gratuites!!
  - code en C : SVM-Light
  - en ligne: GIST
  - sous S-Plus : package `libsvm`
  - sous R : package `kernlab`
  - sous Matlab : SPIDER



Logiciels

- Environ 40 librairies gratuites!!
  - code en C : SVM-Light
  - en ligne: GIST
  - sous S-Plus : package `libsvm`
  - sous R : package `kernlab`
  - sous Matlab : SPIDER



NB : Les logiciels libres sont pour la plupart sous licence GPL

SVM pour la classification binaire

Le problème de classification binaire

- **Données disponibles :** $(x_1, y_1), \ldots, (x_n, y_n)$

Le problème de classification binaire

- **Données disponibles :** $(x_1, y_1), \ldots, (x_n, y_n)$
  - $x_i \in \mathbb{R}^d$ observations

Le problème de classification binaire

- **Données disponibles :** $(x_1, y_1), \ldots, (x_n, y_n)$
  - $x_i \in \mathbb{R}^d$ observations
  - $y_i \in \{-1, +1\}$ classe (ou label ou étiquette) binaire

Le problème de classification binaire

- **Données disponibles :** $(x_1, y_1), \ldots, (x_n, y_n)$
  - $x_i \in \mathbb{R}^d$ observations
  - $y_i \in \{-1, +1\}$ classe (ou label ou étiquette) binaire
- **Problème :** prédiction du label y connaissant x

Le problème de classification binaire

- **Données disponibles :** $(x_1, y_1), \ldots, (x_n, y_n)$
  - $x_i \in \mathbb{R}^d$ observations
  - $y_i \in \{-1, +1\}$ classe (ou label ou étiquette) binaire
- **Problème :** prédiction du label y connaissant x
- **On cherche :** un classifieur $g \ : \ \mathbb{R}^d \to \{-1, +1\}$

Le problème de classification binaire

- **Données disponibles :** $(x_1, y_1), \ldots, (x_n, y_n)$
  - $x_i \in \mathbb{R}^d$ observations
  - $y_i \in \{-1, +1\}$ classe (ou label ou étiquette) binaire
- **Problème :** prédiction du label y connaissant x
- **On cherche :** un classifieur $g : \mathbb{R}^d \to \{-1, +1\}$
- **Question :** trouver un classifieur g qui "généralise" bien.

Le problème de classification binaire

- **Données disponibles :** $(x_1, y_1), \ldots, (x_n, y_n)$
  - $x_i \in \mathbb{R}^d$ observations
  - $y_i \in \{-1, +1\}$ classe (ou label ou étiquette) binaire
- **Problème :** prédiction du label y connaissant x
- **On cherche :** un classifieur $g : \mathbb{R}^d \to \{-1, +1\}$
- **Question :** trouver un classifieur g qui "généralise" bien.
- **Idée :** on choisit g qui "interprète" bien, mais pas trop!

Le problème de classification binaire

- **Données disponibles :** $(x_1, y_1), \ldots, (x_n, y_n)$
  - $x_i \in \mathbb{R}^d$ observations
  - $y_i \in \{-1, +1\}$ classe (ou label ou étiquette) binaire
- **Problème :** prédiction du label y connaissant x
- **On cherche :** un classifieur $g : \mathbb{R}^d \to \{-1, +1\}$
- **Question :** trouver un classifieur g qui "généralise" bien.
- **Idée :** on choisit g qui "interprète" bien, mais pas trop!
- **Concrètement :** on cherche une fonction de décision $f : \mathbb{R}^d \to \mathbb{R}$ et on y associe le classifieur $g = \mathrm{sgn}(f)$

Performance d'un classifieur

- Etant donnée une observation x, un classifieur g réalise une prédiction $g(x)$ à comparer à la classe y.

Performance d'un classifieur

- Etant donnée une observation x, un classifieur g réalise une prédiction $g(x)$ à comparer à la classe y.

Erreur du classifieur = Taux d'observations mal classées

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}_{[g(x_i)\neq y_i]}$$

Performance d'un classifieur

- Etant donnée une observation x, un classifieur g réalise une prédiction $g(x)$ à comparer à la classe y.

Erreur du classifieur = Taux d'observations mal classées

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}_{[g(x_i)\neq y_i]} = \frac{\#\{i \ : \ g(x_i) \neq y_i\}}{n}$$

Cette erreur s'appelle aussi erreur d'apprentissage.

Performance d'un classifieur

- Etant donnée une observation x, un classifieur g réalise une prédiction $g(x)$ à comparer à la classe y.

Erreur du classifieur = Taux d'observations mal classées

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}_{[g(x_i)\neq y_i]} = \frac{\#\{i \ : \ g(x_i) \neq y_i\}}{n}$$

Cette erreur s'appelle aussi erreur d'apprentissage.

- Un classifieur g "interprète" convenablement les données si son erreur d'apprentissage est faible.

Performance d'un classifieur

- Etant donnée une observation x, un classifieur g réalise une prédiction $g(x)$ à comparer à la classe y.

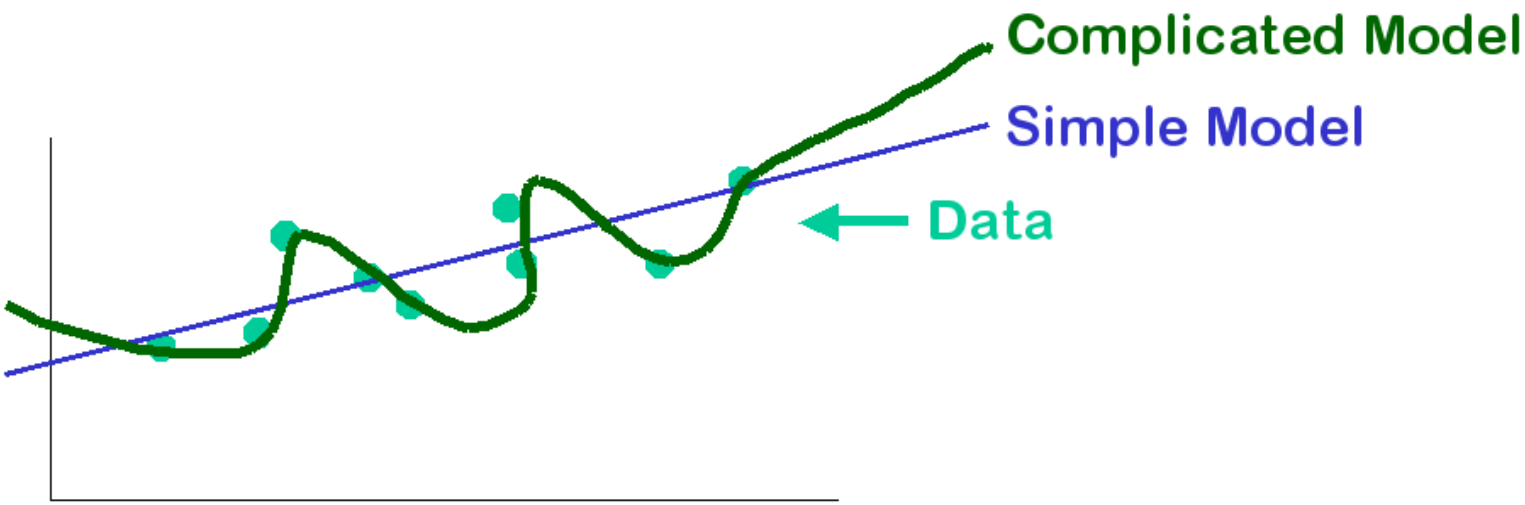
Erreur du classifieur = Taux d'observations mal classées

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}_{[g(x_i)\neq y_i]} = \frac{\#\{i \ : \ g(x_i) \neq y_i\}}{n}$$

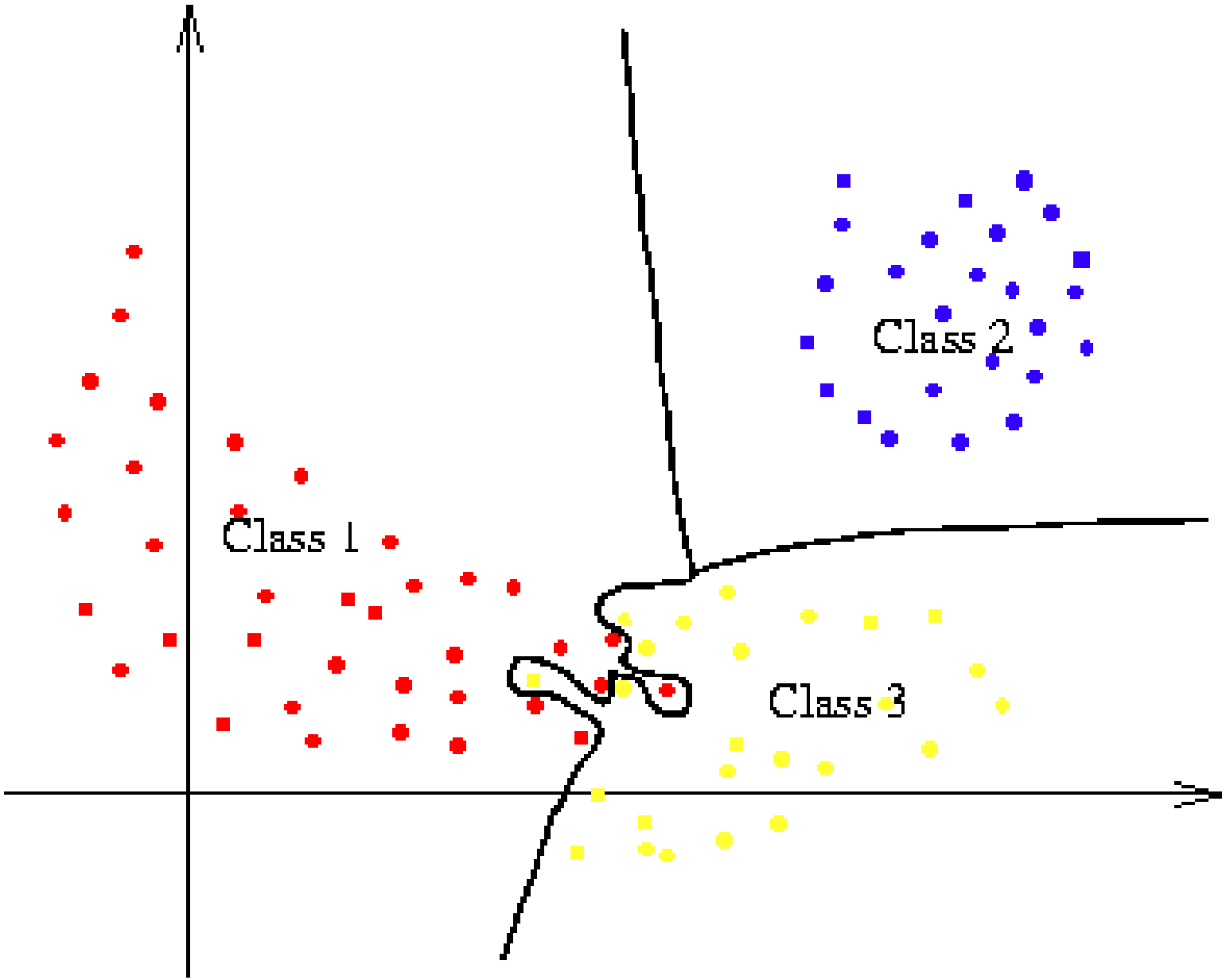
Cette erreur s'appelle aussi erreur d'apprentissage.

- Un classifieur g "interprète" convenablement les données si son erreur d'apprentissage est faible.
- Attention! si on s'intéresse seulement à ce type d'erreur, on risque d'avoir des problèmes...

Overfitting - régression



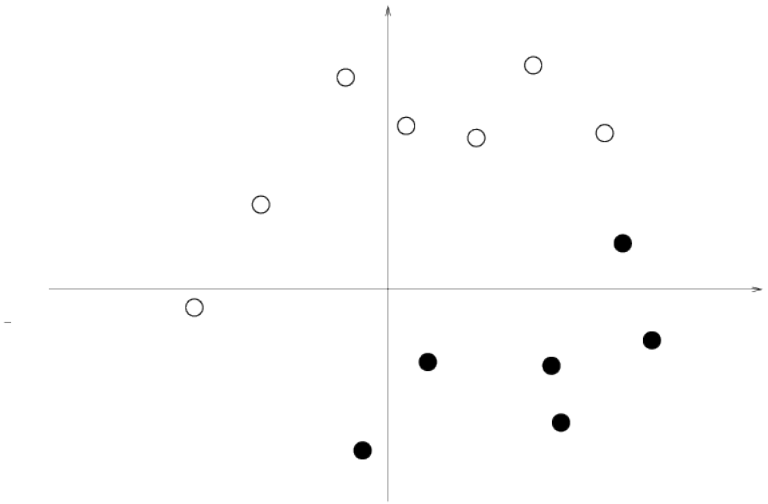
Overfitting - classification



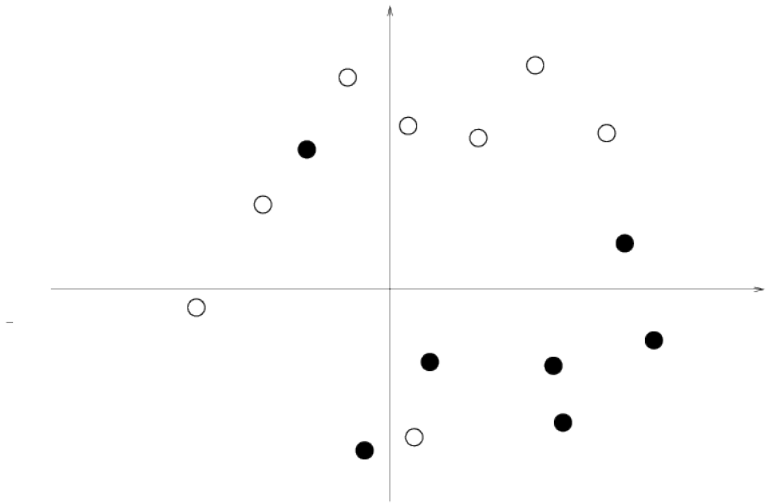
Cas de la classification - Les trois scénarios

1. les populations sont linéairement séparables
2. les populations sont *presque* linéairement séparables
3. les populations ne sont pas linéairement séparables

Séparabilité linéaire



Scénario 1



Scénario 2

Notations

- **Produit scalaire** entre deux vecteurs sur $\mathbb{R}^d$:

$$\forall u,\ v \in \mathbb{R}^d\ , \quad < u, v >= u^T v = \sum_{j=1}^{d} u_j v_j$$

où $u = (u_1, \ldots, u_d)^T$ et $v = (v_1, \ldots, v_d)^T$

Notations

- **Produit scalaire** entre deux vecteurs sur $\mathbb{R}^d$:

$$\forall u,\ v \in \mathbb{R}^d\ , \quad < u, v >= u^T v = \sum_{j=1}^{d} u_j v_j$$

où $u = (u_1, \ldots, u_d)^T$ et $v = (v_1, \ldots, v_d)^T$

- **Norme euclidienne** sur $\mathbb{R}^d$:

$$\forall u \in \mathbb{R}^d,\ \|u\| = \sqrt{< u, u >} = \sqrt{\sum_{j=1}^{d} u_j^2}$$

Séparateurs linéaires

- **Forme des fonctions de décision :**

$$f(x) = b + < \beta, x >$$

où $b \in \mathbb{R}$, $\beta \in \mathbb{R}^d$.

Séparateurs linéaires

- **Forme des fonctions de décision :**

$$f(x) = b + < \beta, x >$$

où $b \in \mathbb{R}$, $\beta \in \mathbb{R}^d$.

- L'équation $f(x) = 0$ définit un **hyperplan séparateur** H dans $\mathbb{R}^d$

Séparateurs linéaires

- **Forme des fonctions de décision :**

$$f(x) = b + < \beta, x >$$

où $b \in \mathbb{R}$, $\beta \in \mathbb{R}^d$.

- L'équation $f(x) = 0$ définit un **hyperplan séparateur** H dans $\mathbb{R}^d$
- **Classifieur associé :**

$$\forall x \in \mathbb{R}^d \quad g_f(x) = \begin{cases} +1 & \text{si } f(x) > 0 \\ \\ -1 & \text{si } f(x) \leq 0 \end{cases}$$

Quelques propriétés

1. $\beta^* = \dfrac{\beta}{\|\beta\|}$ est le vecteur normal à H

Quelques propriétés

1. $\beta^* = \dfrac{\beta}{\|\beta\|}$ est le vecteur normal à H
2. $\forall x_0 \in H, \quad < \beta, x_0 >= -b$

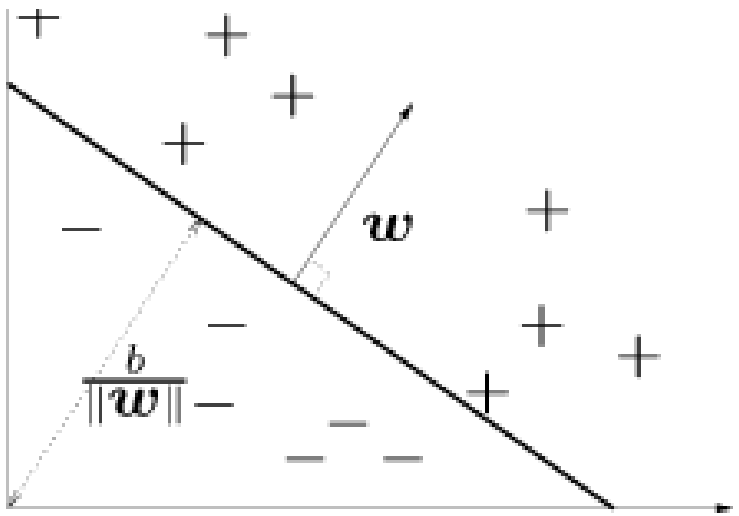
Quelques propriétés

1. $\beta^* = \dfrac{\beta}{\|\beta\|}$ est le vecteur normal à H
2. $\forall x_0 \in H, \quad < \beta, x_0 >= -b$
3. la distance signée (éventuellement négative !) d'un point $x \in \mathbb{R}^d$ à H est donnée par

$$d(x, H) =< \beta^*, x - x_0 >= \frac{1}{\|\beta\|}\big(b + < \beta, x >\big)$$

où $x_0 \in H$

Scénario 1 - Une figure



A separating hyperplane $(\boldsymbol{w}, b) \in \mathbb{R}^n \times \mathbb{R}$ for a 2D training set.

Attention! ici $w = \beta$...

Algorithme du perceptron (Rosenblatt, 1958)

Perceptron - version simplifiée $b = 0$

Génère une suite $\beta_0, \ldots, \beta_n$ de valeurs pour β

1. **Initialisation** - $\beta_0 = 0$

Algorithme du perceptron (Rosenblatt, 1958)

Perceptron - version simplifiée $b = 0$

Génère une suite $\beta_0, \ldots, \beta_n$ de valeurs pour β

1. **Initialisation** - $\beta_0 = 0$
2. **Etape i** - on considère le couple (x_i, y_i) et on regarde s'il est correctement classé ou non

$$\beta_i = \begin{cases} \beta_{i-1} & \text{si } y_i \cdot < \beta_{i-1}, x_i > > 0 \\ \beta_{i-1} + y_i x_i & \text{si } y_i \cdot < \beta_{i-1}, x_i > \leq 0 \end{cases}$$

Algorithme du perceptron général

Perceptron - version générale

- **Paramètres :**

Algorithme du perceptron général

Perceptron - version générale

- **Paramètres :**
  - taux d'apprentissage η

Algorithme du perceptron général

Perceptron - version générale

- **Paramètres :**
  - taux d'apprentissage η
  - rayon des observations $R = \max_{1 \leq i \leq n} \|x_i\|$

Algorithme du perceptron général

Perceptron - version générale

- **Paramètres :**
  - taux d'apprentissage η
  - rayon des observations $R = \max_{1 \leq i \leq n} \|x_i\|$
- **Algorithme :**

Algorithme du perceptron général

Perceptron - version générale

- **Paramètres :**
  - taux d'apprentissage η
  - rayon des observations $R = \max_{1 \leq i \leq n} \|x_i\|$
- **Algorithme :**
  1. **Initialisation** - $\beta_0 = 0,\ b_0 = 0$

Algorithme du perceptron général

Perceptron - version générale

- **Paramètres :**
  - taux d'apprentissage η
  - rayon des observations $R = \max_{1 \leq i \leq n} \|x_i\|$

- **Algorithme :**
  1. **Initialisation** - $\beta_0 = 0,\ b_0 = 0$
  2. **Etape i** - si (x_i, y_i) est mal classé par l'hyperplan (b_{i-1}, β_{i-1}), alors:

$$\begin{aligned} \beta_i &= \beta_{i-1} + \eta y_i x_i \\ b_i &= b_{i-1} + \eta y_i^2 R^2 \end{aligned}$$

sinon $\beta_i = \beta_{i-1},\ b_i = b_{i-1}$

Propriétés du perceptron

Théorème de Novikoff

Si les populations sont linéairement séparables alors l'algorithme du perceptron converge en un nombre fini $T \leq n$ d'étapes où:

$$T \leq \frac{2R^2}{M^2}$$

avec $M = \min_{1 \leq i \leq n} \{y_i d(x_i, H^*)\}$ pour un certain séparateur H^*.

Propriétés du perceptron

Théorème de Novikoff

Si les populations sont linéairement séparables alors l'algorithme du perceptron converge en un nombre fini $T \leq n$ d'étapes où:

$$T \leq \frac{2R^2}{M^2}$$

avec $M = \min_{1 \leq i \leq n} \{y_i d(x_i, H^*)\}$ pour un certain séparateur H^*.

- **Défaut du perceptron :** mauvaise généralisation

Propriétés du perceptron

Théorème de Novikoff

Si les populations sont linéairement séparables alors l'algorithme du perceptron converge en un nombre fini $T \leq n$ d'étapes où:

$$T \leq \frac{2R^2}{M^2}$$

avec $M = \min_{1 \leq i \leq n} \{y_i d(x_i, H^*)\}$ pour un certain séparateur H^*.

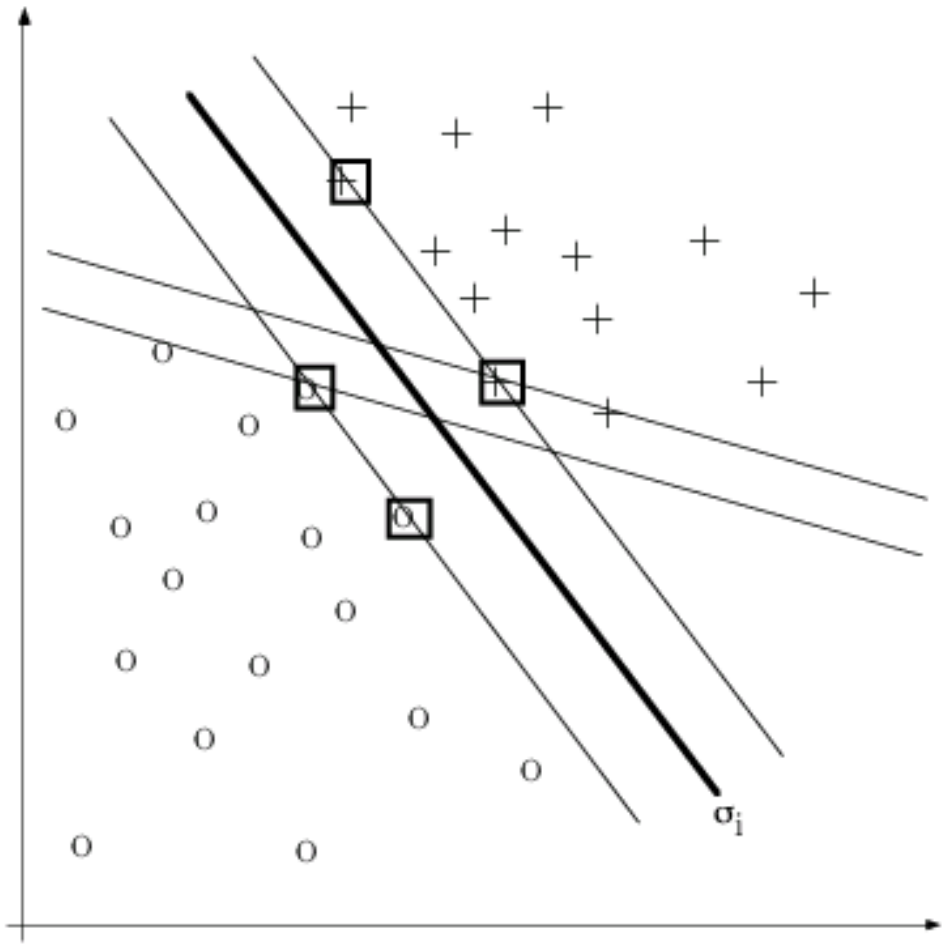
- **Défaut du perceptron :** mauvaise généralisation
- **Vertu du perceptron :** algorithme séquentiel (online)

Scénario 1 - hyperplan à bonne généralisation

Question: hyperplan se trouvant à distance maximale de chaque population ?

Scénario 1 - hyperplan à bonne généralisation

Question: hyperplan se trouvant à distance maximale de chaque population ?



Hyperplan à marges optimales (Vapnik-Chervonenkis, 1964)

Problème d'optimisation

$$\max_{\beta \in \mathbb{R}^d,\ b \in \mathbb{R}} \quad M$$

sous les contraintes:

$$\forall i = 1, \ldots, n \ , \quad y_i \cdot d(x_i, H) \geq M$$

On rappelle:

$$d(x_i, H) = \frac{1}{\|\beta\|} \left(b + < \beta, x_i > \right)$$

Problème quadratique

Contraintes:

$$\forall i = 1, \ldots, n \ , \quad y_i \cdot \frac{1}{\|\beta\|}\left(b + < \beta, x_i >\right) \geq M$$

On peut très bien poser: $M = 1/\|\beta\|$

Formulation équivalente

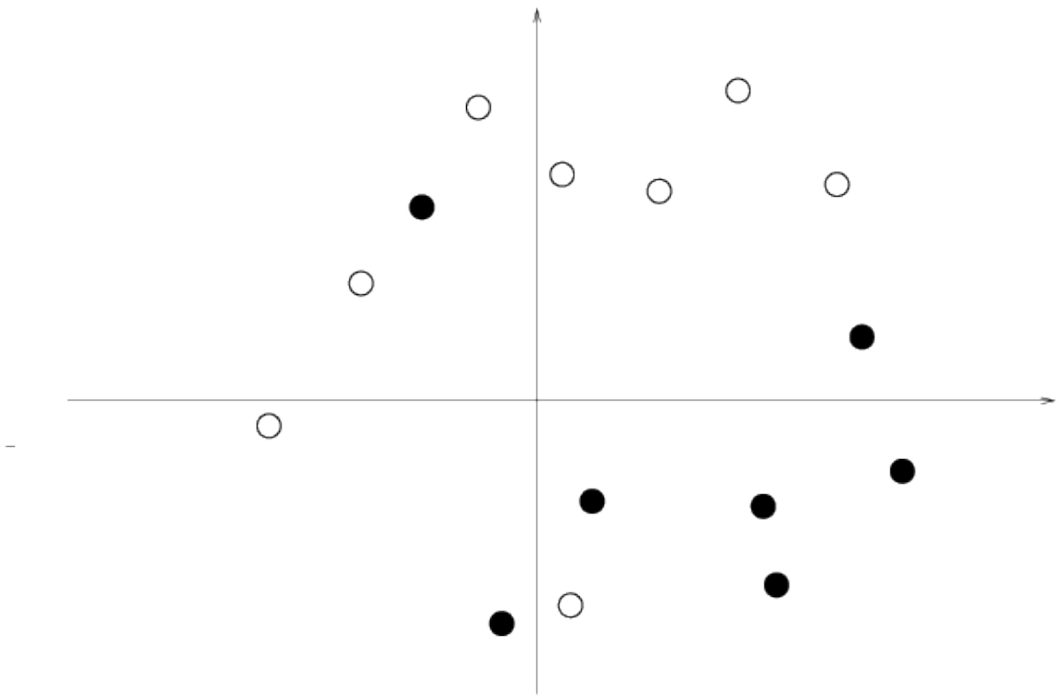
$$\min_{\beta, b} \frac{1}{2}\|\beta\|^2$$

sous les contraintes:

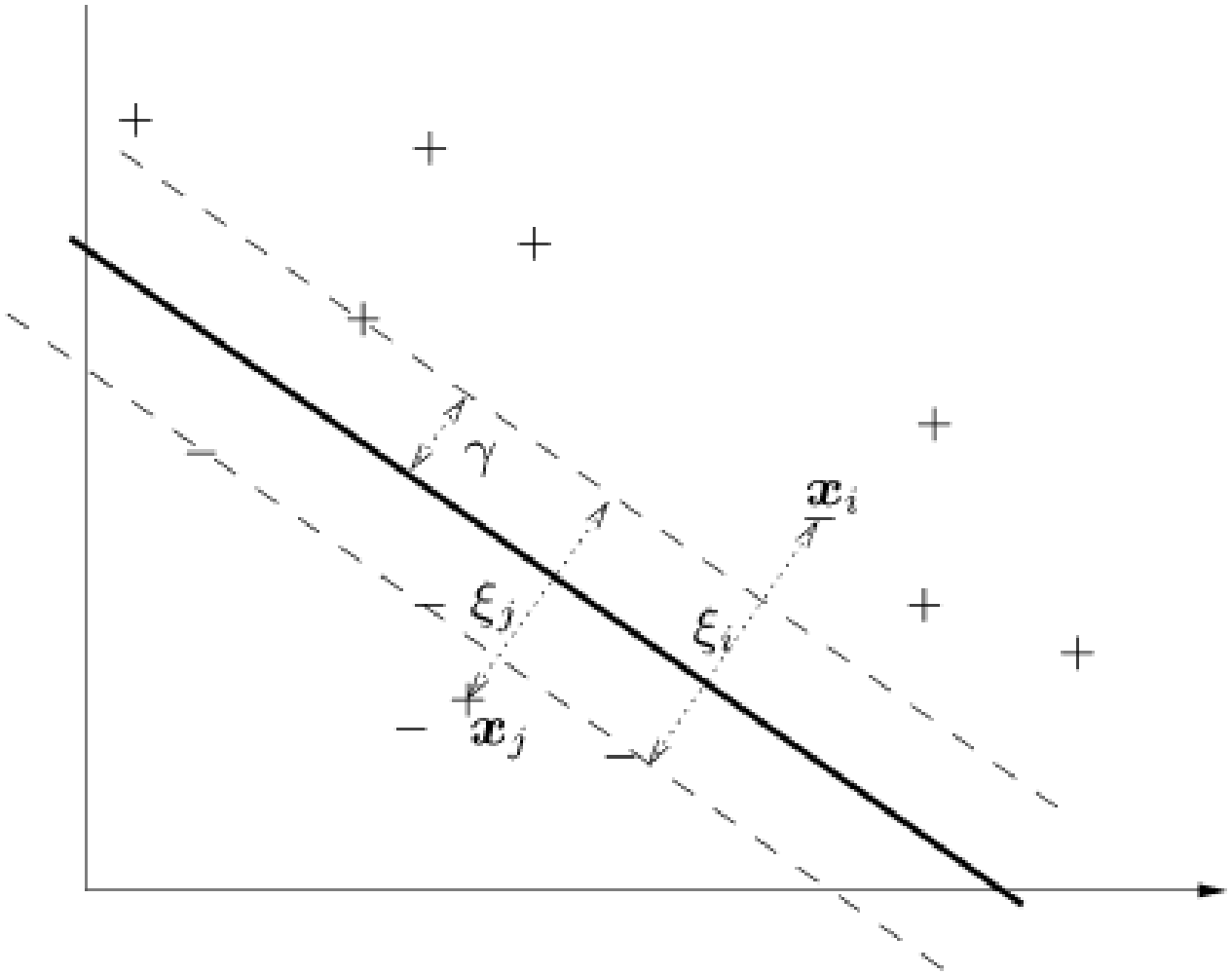
$$\forall i = 1, \ldots, n \ , \quad y_i \cdot \left(b + < \beta, x_i >\right) \geq 1$$

Scénario 2 - Populations presque linéairement séparables

Idée : la "bonne" séparation est linéaire mais certaines observations ont des labels bruités.



Scénario 2 - Variables "ressorts"



Scénario 2 - Variables "ressorts" (suite)

On introduit n variables supplémentaires ("slacks" ou "ressorts"):
$\xi = (\xi_1, \ldots, \xi_n)$ avec $\xi_i \geq 0, \forall i$

Nouveau problème d'optimisation

$$\min_{\beta, b} \frac{1}{2}\|\beta\|^2$$

sous les contraintes:

$$\begin{aligned} \forall i = 1, \ldots, n\ , \quad y_i \cdot \left(b + < \beta, x_i >\right) &\geq 1 - \xi_i \\ \xi_i &\geq 0 \\ \sum_{i=1}^{n} \xi_i &\leq \Xi \end{aligned}$$

Formulation lagrangienne I

Formulation lagrangienne I

$$\min_{\beta,b} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n} \xi_i$$

sous les contraintes:

$$\begin{aligned} \forall i = 1, \ldots, n\ , \quad \xi_i &\geq 0 \\ \xi_i &\geq 1 - [y_i \cdot \left(b + < \beta, x_i >\right)] \end{aligned}$$

Formulation lagrangienne II

Multiplicateurs de Lagrange: $\alpha = (\alpha_1, \ldots, \alpha_n)$, $\mu = (\mu_1, \ldots, \mu_n)$

Formulation lagrangienne II

$$\min_{\beta,b,\xi} \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left(y_i \cdot \left(b + <\beta, x_i>\right) - (1-\xi_i)\right) + \sum_{i=1}^{n}\mu_i\xi_i$$

Conditions du premier ordre (gradient nul)

$$\begin{aligned} \beta &= \sum_{i=1}^{n}\alpha_i y_i x_i \\ \sum_{i=1}^{n}\alpha_i y_i &= 0 \\ \forall i = 1, \ldots, n\ , \qquad \alpha_i &= C + \mu_i \end{aligned}$$

Formulation duale

Formulation duale

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j < x_i, x_j >$$

sous les contraintes:

$$\begin{aligned} \forall i = 1, \ldots, n \ , \qquad 0 &\leq \alpha_i \leq C \\ \sum_{i=1}^{n} \alpha_i y_i &= 0 \end{aligned}$$

On note $\hat{\alpha} = (\hat{\alpha}_1, \ldots, \hat{\alpha}_n)$ la solution de ce problème.

Conditions de Karush-Kuhn-Tucker

Conditions de Karush-Kuhn-Tucker

$$\forall i = 1, \ldots, n \ , \qquad \begin{aligned} \alpha_i \big(y_i \cdot f(x_i) - (1 - \xi_i)\big) &= 0 \\ y_i \cdot f(x_i) - (1 - \xi_i) &\geq 0 \\ \alpha_i + \mu_i &= C \\ \mu_i \xi_i &= 0 \\ \beta &= \sum_{i=1}^{n} \alpha_i y_i x_i \\ \sum_{i=1}^{n} \alpha_i y_i &= 0 \end{aligned}$$

Interprétation

Lien entre les coefficients et la position des observations

- si $\hat{\alpha}_i = 0$ alors $y_i \cdot f(x_i) \geq 1 \Rightarrow$ le point x_i est bien classé car $\mu_i = C > 0$ et on a $\xi_i = 0$

Interprétation

Lien entre les coefficients et la position des observations

- si $\hat{\alpha}_i = 0$ alors $y_i \cdot f(x_i) \geq 1 \Rightarrow$ le point x_i est bien classé
  car $\mu_i = C > 0$ et on a $\xi_i = 0$

- si $0 < \hat{\alpha}_i < C$ alors $y_i \cdot f(x_i) = 1 \Rightarrow$ le point x_i est sur la frontière de la marge
  car $\mu_i > 0$ et $\xi_i = 0$

Interprétation

Lien entre les coefficients et la position des observations

- si $\hat{\alpha}_i = 0$ alors $y_i \cdot f(x_i) \geq 1 \Rightarrow$ le point x_i est bien classé
  car $\mu_i = C > 0$ et on a $\xi_i = 0$
- si $0 < \hat{\alpha}_i < C$ alors $y_i \cdot f(x_i) = 1 \Rightarrow$ le point x_i est sur la frontière de la marge
  car $\mu_i > 0$ et $\xi_i = 0$
- si $\hat{\alpha}_i = C$ alors $y_i \cdot f(x_i) \leq 1 \Rightarrow$ le point x_i dépasse la frontière de la marge
  car $\mu_i = 0$ et donc $\xi_i \geq 0$

Interprétation

Lien entre les coefficients et la position des observations

- si $\hat{\alpha}_i = 0$ alors $y_i \cdot f(x_i) \geq 1 \Rightarrow$ le point x_i est bien classé
  car $\mu_i = C > 0$ et on a $\xi_i = 0$
- si $0 < \hat{\alpha}_i < C$ alors $y_i \cdot f(x_i) = 1 \Rightarrow$ le point x_i est sur la frontière de la marge
  car $\mu_i > 0$ et $\xi_i = 0$
- si $\hat{\alpha}_i = C$ alors $y_i \cdot f(x_i) \leq 1 \Rightarrow$ le point x_i dépasse la frontière de la marge
  car $\mu_i = 0$ et donc $\xi_i \geq 0$

Interprétation

Lien entre les coefficients et la position des observations

- si $\hat{\alpha}_i = 0$ alors $y_i \cdot f(x_i) \geq 1 \Rightarrow$ le point x_i est bien classé
  car $\mu_i = C > 0$ et on a $\xi_i = 0$
- si $0 < \hat{\alpha}_i < C$ alors $y_i \cdot f(x_i) = 1 \Rightarrow$ le point x_i est sur la frontière de la marge
  car $\mu_i > 0$ et $\xi_i = 0$
- si $\hat{\alpha}_i = C$ alors $y_i \cdot f(x_i) \leq 1 \Rightarrow$ le point x_i dépasse la frontière de la marge
  car $\mu_i = 0$ et donc $\xi_i \geq 0$

Phénomène remarquable!

En pratique, beaucoup de $\hat{\alpha}_i$ sont nuls!

Solution du problème

Définition

Les $\hat{\alpha}_i \neq 0$ correspondent aux vecteurs de support. On note I l'ensemble des indices parmi $\{1, \ldots, n\}$ correspondants.

Solution du problème

Définition

Les $\hat{\alpha}_i \neq 0$ correspondent aux vecteurs de support. On note I l'ensemble des indices parmi $\{1, \ldots, n\}$ correspondants.

Représentation de la solution

Fonction de décision:

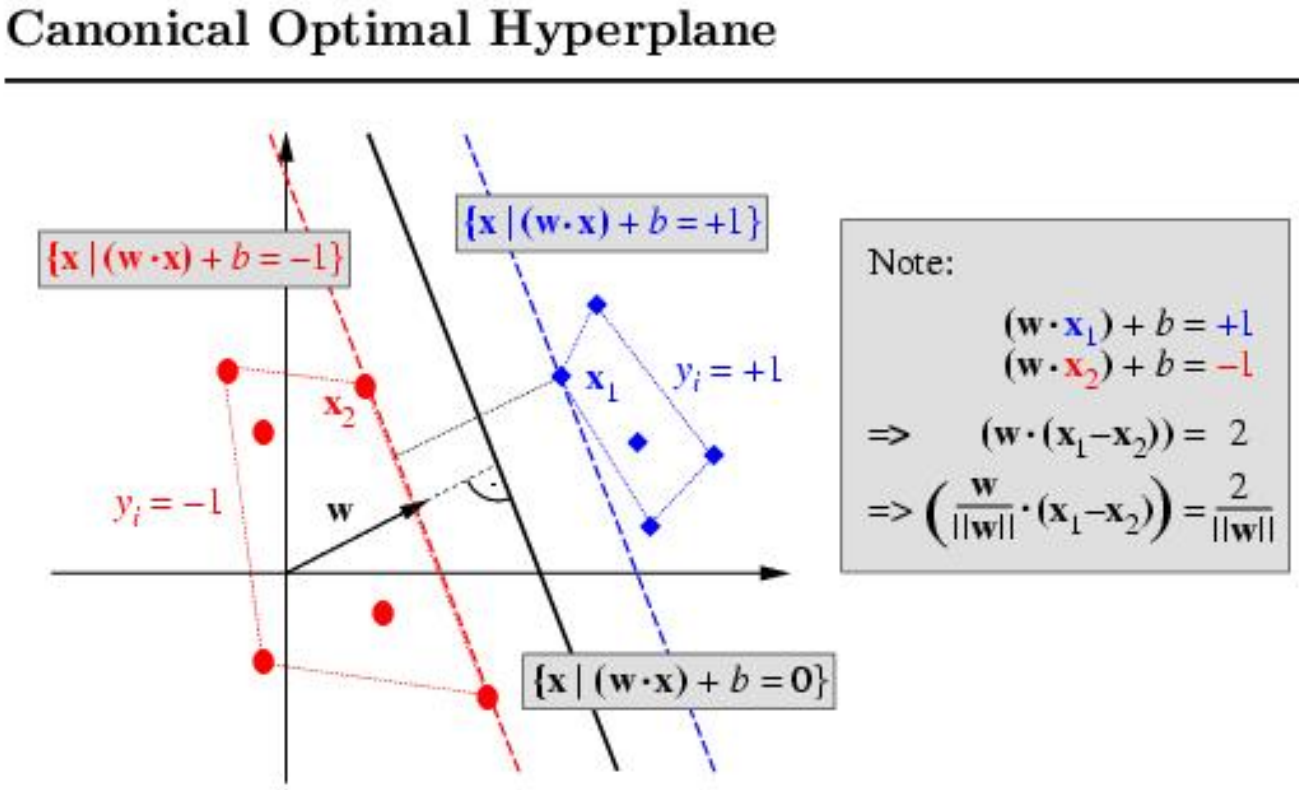
$$\hat{f}(x) = \hat{b} + \sum_{i \in I} \hat{\alpha}_i y_i < x_i, x >$$

où :

$$\hat{\beta} = \sum_{i \in I} \hat{\alpha}_i y_i x_i \ , \quad I = \{i \ : \ \hat{\alpha}_i \neq 0\}$$

$$\hat{b} = y_j - \sum_{i \in I} \hat{\alpha}_i y_i < x_i, x_j > \ , \quad \text{pour un certain } j \in I$$

Vecteurs de support



⇒ Représentation parcimonieuse (”sparse”) des SVM

Scénario 3 - remarques préliminaires

- La plupart des problèmes de classification relèvent de séparations non-linéaires

Scénario 3 - remarques préliminaires

- La plupart des problèmes de classification relèvent de séparations non-linéaires
- L'algorithme de construction de l'hyperplan à marges optimales ne fait intervenir les observations que sous la forme des produits scalaires $< x_i, x_j >$ pour tout i, j

Scénario 3 - remarques préliminaires

- La plupart des problèmes de classification relèvent de séparations non-linéaires
- L'algorithme de construction de l'hyperplan à marges optimales ne fait intervenir les observations que sous la forme des produits scalaires $< x_i, x_j >$ pour tout i, j
- La fonction de décision dépend du produit scalaire entre le nouveau point x et les vecteurs de support x_i

L'astuce du noyau

- **Astuce du noyau ("kernel trick") :** l'algorithme de construction de l'hyperplan à marges optimales ne dépend des observations qu'au travers des coefficients de la matrice de Gram

$$K = (< x_i, x_j >)_{1 \leq i,j \leq n}$$

L'astuce du noyau

- **Astuce du noyau ("kernel trick") :** l'algorithme de construction de l'hyperplan à marges optimales ne dépend des observations qu'au travers des coefficients de la matrice de Gram

$$K = (< x_i, x_j >)_{1 \leq i,j \leq n}$$

- **Méthodes à noyaux :** on remplace le produit scalaire canonique par un noyau positif

$$k \ : \ \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$

et alors K devient la matrice des $k(x_i, x_j)$.

Support Vector Machines

Formulation duale

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

sous les contraintes:

$$\begin{aligned} \forall i = 1, \ldots, n \, , \qquad 0 &\leq \alpha_i \leq C \\ \sum_{i=1}^{n} \alpha_i y_i &= 0 \end{aligned}$$

Support Vector Machines

Formulation duale

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

sous les contraintes:

$$\begin{aligned} \forall i = 1, \ldots, n\ , \qquad 0 &\leq \alpha_i \leq C \\ \sum_{i=1}^{n} \alpha_i y_i &= 0 \end{aligned}$$

Fonction de décision :

$$\hat{f}(x) = \hat{b} + \sum_{i \in I} \hat{\alpha}_i k(x_i, x)$$

avec I indices des vecteurs de support.

Exemples de noyaux

- noyaux polynomiaux

$$\begin{aligned} k_r(x, x') &= (< x, x' >)^r \\ k_{r,c}(x, x') &= (< x, x' > + c)^r \end{aligned}$$

Exemples de noyaux

- noyaux polynomiaux

$$\begin{aligned} k_r(x, x') &= (< x, x' >)^r \\ k_{r,c}(x, x') &= (< x, x' > + c)^r \end{aligned}$$

- noyau à fonctions de base radiales gaussiennes (RBF)

$$k_\sigma(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

Exemples de noyaux

- noyaux polynomiaux

$$\begin{aligned} k_r(x, x') &= (< x, x' >)^r \\ k_{r,c}(x, x') &= (< x, x' > + c)^r \end{aligned}$$

- noyau à fonctions de base radiales gaussiennes (RBF)

$$k_\sigma(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

- noyau sigmoïde (réseau de neurones)

$$k_{\kappa,\theta}(x, x') = \tanh\left(\kappa < x, x' > + \theta\right)$$

Problèmes pratiques

- Sélection d'un noyau

Problèmes pratiques

- Sélection d'un noyau
- Réglage des paramètres : utilisation d'une base de validation

Problèmes pratiques

- Sélection d'un noyau
- Réglage des paramètres : utilisation d'une base de validation
- Mesures de performances : erreur sur la base de test

Problèmes pratiques

- Sélection d'un noyau
- Réglage des paramètres : utilisation d'une base de validation
- Mesures de performances : erreur sur la base de test
- Comparaison à d'autres méthodes

Problèmes pratiques

- Sélection d'un noyau
- Réglage des paramètres : utilisation d'une base de validation
- Mesures de performances : erreur sur la base de test
- Comparaison à d'autres méthodes
- Extensions:

Problèmes pratiques

- Sélection d'un noyau
- Réglage des paramètres : utilisation d'une base de validation
- Mesures de performances : erreur sur la base de test
- Comparaison à d'autres méthodes
- Extensions:
  - problème de classification à plus de deux classes

Problèmes pratiques

- Sélection d'un noyau
- Réglage des paramètres : utilisation d'une base de validation
- Mesures de performances : erreur sur la base de test
- Comparaison à d'autres méthodes
- Extensions:
  - problème de classification à plus de deux classes
  - cas de populations très disproportionnées

Problèmes multi-classes

Soit ℓ le nombre de classes

- **Un-contre-tous**
  $(\ell - 1)$ classifieurs binaires + vote

Problèmes multi-classes

Soit ℓ le nombre de classes

- **Un-contre-tous**
  $(\ell - 1)$ classifieurs binaires + vote
- **Un-contre-un**
  C_ℓ^2 classifieurs binaires + vote

Problèmes multi-classes

Soit ℓ le nombre de classes

- **Un-contre-tous**
  $(\ell - 1)$ classifieurs binaires + vote
- **Un-contre-un**
  C_ℓ^2 classifieurs binaires + vote
- **Fusion de classes** (error-correcting output codes)
  Plusieurs classifieurs binaires + règle de superposition

Problèmes multi-classes

Soit ℓ le nombre de classes

- **Un-contre-tous**
  $(\ell - 1)$ classifieurs binaires + vote
- **Un-contre-un**
  C_ℓ^2 classifieurs binaires + vote
- **Fusion de classes** (error-correcting output codes)
  Plusieurs classifieurs binaires + règle de superposition
- **Critères d'optimisation multi-classes**
  Algorithme SVM modifié + règle de classification :

$$g(x) = \arg\min_{1 \leq j \leq \ell} \left(b_j + \sum_{i : y_i = j} \alpha_i k(x_i, x) \right) \in \{1, \ldots, \ell\}$$

Exemples I - données artificielles

Utilisation du logiciel Spider

- Développé au Max Planck Institute (Tübingen)

Utilisation du logiciel Spider

- Développé au Max Planck Institute (Tübingen)
- Disponible en ligne :

Utilisation du logiciel Spider

- Développé au Max Planck Institute (Tübingen)
- Disponible en ligne :

Utilisation du logiciel Spider

- Développé au Max Planck Institute (Tübingen)
- Disponible en ligne :

`http://www.kyb.tuebingen.mpg.de/bs/people/spider/main.html`

Utilisation du logiciel Spider

- Développé au Max Planck Institute (Tübingen)
- Disponible en ligne :

`http://www.kyb.tuebingen.mpg.de/bs/people/spider/main.html`

Premiers pas :

```
X = randn(5, 10);
Y = [1,1,1,-1,-1]';
d = data(X,Y);
a=svm;
[tr, a] = train(a,d)
```

Effet des paramètres C et σ

Données = réalisations de deux gaussiennes bivariées

Script :

```
m = 20; d = 1; s = 1;

x1 = [randn(m,1)*s-d randn(m,1)*s-d];
x2 = [randn(m,1)*s+d randn(m,1)*s+d];
d = data([x1;x2], [ones(m,1) ; -ones(m,1)]);

a = svm; sigma = 1; a.C = 1;
a.child = kernel('rbf', sigma);
[r a] = train(a,d);
plot(a)
```

Eléments de théorie pour les SVM

Formalisme probabiliste pour la classification binaire

- **Observation:** $X \in \mathbb{R}^d$ (ou $\mathcal{X}$), loi μ

Formalisme probabiliste pour la classification binaire

- **Observation:** $X \in \mathbb{R}^d$ (ou $\mathcal{X}$), loi μ
- **Label/Classe:** $Y \in \{-1, +1\}$

Formalisme probabiliste pour la classification binaire

- **Observation:** $X \in \mathbb{R}^d$ (ou $\mathcal{X}$), loi μ
- **Label/Classe:** $Y \in \{-1, +1\}$
- **Fonction de régression :**

$$\eta(x) = \mathbb{P}\{Y = 1 | X = x\}$$

Formalisme probabiliste pour la classification binaire

- **Observation:** $X \in \mathbb{R}^d$ (ou $\mathcal{X}$), loi μ
- **Label/Classe:** $Y \in \{-1, +1\}$
- **Fonction de régression :**

$$\eta(x) = \mathbb{P}\{Y = 1 | X = x\}$$

- **Echantillon :** données indépendantes et de même loi

$$D_n = \{(X_1, Y_1), ..., (X_n, Y_n)\}$$

Formalisme probabiliste pour la classification binaire

- **Observation:** $X \in \mathbb{R}^d$ (ou $\mathcal{X}$), loi μ
- **Label/Classe:** $Y \in \{-1, +1\}$
- **Fonction de régression :**

$$\eta(x) = \mathbb{P}\{Y = 1 | X = x\}$$

- **Echantillon :** données indépendantes et de même loi

$$D_n = \{(X_1, Y_1), ..., (X_n, Y_n)\}$$

- **Classifieur :**

$$\text{fait une prédiction } g(X) \in \{-1, +1\}$$

Notion de risque

- **Une mesure d'erreur :**

$$L(g) = \mathbb{P}\{Y \cdot g(X) < 0\} = \mathbb{E}\left(\mathbb{I}_{[Y \cdot g(X) < 0]}\right)$$

Notion de risque

- **Une mesure d'erreur :**

$$L(g) = \mathbb{P}\{Y \cdot g(X) < 0\} = \mathbb{E}\left(\mathbb{I}_{[Y \cdot g(X) < 0]}\right)$$

- **Classifieur et erreur de Bayes :**

$$\begin{aligned} g^* &= \arg\min_g L(g) = \operatorname{sgn}\left(\eta - \frac{1}{2}\right) \\ L^* &= L(g^*) \end{aligned}$$

Notion de risque

- **Une mesure d'erreur :**

$$L(g) = \mathbb{P}\{Y \cdot g(X) < 0\} = \mathbb{E}\left(\mathbb{I}_{[Y \cdot g(X) < 0]}\right)$$

- **Classifieur et erreur de Bayes :**

$$\begin{aligned} g^* &= \arg\min_g L(g) = \operatorname{sgn}\left(\eta - \frac{1}{2}\right) \\ L^* &= L(g^*) \end{aligned}$$

- **Critère empirique :**

$$\hat{L}_n(g) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}_{[Y_i \cdot g(X_i) < 0]}$$

Minimisation du risque empirique (ERM)

La théorie de Vapnik ('95 $\rightarrow$) a permis de développer des stratégies consistantes pour la classification de données en grande dimension.

Inconvénients de l'ERM

- algorithmique : problème NP-difficile .

Minimisation du risque empirique (ERM)

La théorie de Vapnik ('95 $\rightarrow$) a permis de développer des stratégies consistantes pour la classification de données en grande dimension.

Inconvénients de l'ERM

- algorithmique : problème NP-difficile .
- contrôle de la complexité : propriété de Glivenko-Cantelli pour éviter le surapprentissage (overfitting).

Minimisation du risque empirique (ERM)

La théorie de Vapnik ('95 $\rightarrow$) a permis de développer des stratégies consistantes pour la classification de données en grande dimension.

Inconvénients de l'ERM

- algorithmique : problème NP-difficile .
- contrôle de la complexité : propriété de Glivenko-Cantelli pour éviter le surapprentissage (overfitting).

Minimisation du risque empirique (ERM)

La théorie de Vapnik ('95 $\rightarrow$) a permis de développer des stratégies consistantes pour la classification de données en grande dimension.

Inconvénients de l'ERM

- algorithmique : problème NP-difficile .
- contrôle de la complexité : propriété de Glivenko-Cantelli pour éviter le surapprentissage (overfitting).

Mais :

Les méthodes efficaces construisent les estimateurs dans des classes massives!

Algorithmes efficaces

1. **Support Vector Machines** - Vapnik (1995)

Algorithmes efficaces

1. **Support Vector Machines** - Vapnik (1995)
   - Kernel trick : envoyer les données dans un espace de Hilbert où les données soient (presque) linéairement séparables

Algorithmes efficaces

1. **Support Vector Machines** - Vapnik (1995)
   - Kernel trick : envoyer les données dans un espace de Hilbert où les données soient (presque) linéairement séparables
   - Hyperplan à marge maximale : optimization convexe sous contraintes quadratiques

Algorithmes efficaces

1. **Support Vector Machines** - Vapnik (1995)
   - Kernel trick : envoyer les données dans un espace de Hilbert où les données soient (presque) linéairement séparables
   - Hyperplan à marge maximale : optimization convexe sous contraintes quadratiques
2. **Boosting** - Freund (1990) - Freund, Schapire (1996)

Algorithmes efficaces

1. **Support Vector Machines** - Vapnik (1995)
   - Kernel trick : envoyer les données dans un espace de Hilbert où les données soient (presque) linéairement séparables
   - Hyperplan à marge maximale : optimization convexe sous contraintes quadratiques
2. **Boosting** - Freund (1990) - Freund, Schapire (1996)
   - on commence avec une classe $\mathcal{H}$ de classifieurs simples

Algorithmes efficaces

1. **Support Vector Machines** - Vapnik (1995)
   - Kernel trick : envoyer les données dans un espace de Hilbert où les données soient (presque) linéairement séparables
   - Hyperplan à marge maximale : optimization convexe sous contraintes quadratiques

2. **Boosting** - Freund (1990) - Freund, Schapire (1996)
   - on commence avec une classe $\mathcal{H}$ de classifieurs simples
   - puis on construit itérativement une combinaison linéaire de classifieurs simples qui fait décroître l'erreur empirique

Algorithmes efficaces

1. **Support Vector Machines** - Vapnik (1995)
   - Kernel trick : envoyer les données dans un espace de Hilbert où les données soient (presque) linéairement séparables
   - Hyperplan à marge maximale : optimization convexe sous contraintes quadratiques

2. **Boosting** - Freund (1990) - Freund, Schapire (1996)
   - on commence avec une classe $\mathcal{H}$ de classifieurs simples
   - puis on construit itérativement une combinaison linéaire de classifieurs simples qui fait décroître l'erreur empirique

Algorithmes efficaces

1. **Support Vector Machines** - Vapnik (1995)
   - Kernel trick : envoyer les données dans un espace de Hilbert où les données soient (presque) linéairement séparables
   - Hyperplan à marge maximale : optimization convexe sous contraintes quadratiques

2. **Boosting** - Freund (1990) - Freund, Schapire (1996)
   - on commence avec une classe $\mathcal{H}$ de classifieurs simples
   - puis on construit itérativement une combinaison linéaire de classifieurs simples qui fait décroître l'erreur empirique

"Boosting is the best off-the-shelf classifier in the world" - Breiman, 1996.

Caractéristique commune

Si on fait abstraction:

1. des intuitions géometriques

Caractéristique commune

Si on fait abstraction:

1. des intuitions géometriques
2. de la dynamique particulière de chaque algorithme, alors:

Caractéristique commune

Si on fait abstraction:

1. des intuitions géometriques
2. de la dynamique particulière de chaque algorithme, alors:

Caractéristique commune

Si on fait abstraction:

1. des intuitions géometriques
2. de la dynamique particulière de chaque algorithme, alors:

Boosting et SVM peuvent s'envisager comme des:

procédures de minimisation d'un risque convexe pénalisé dans des espaces fonctionnels massifs.

Caractéristique commune

Si on fait abstraction:

1. des intuitions géometriques
2. de la dynamique particulière de chaque algorithme, alors:

Boosting et SVM peuvent s'envisager comme des:

procédures de minimisation d'un risque convexe pénalisé dans des espaces fonctionnels massifs.

Elles sont caractérisées par

- des classes d'estimateurs différentes,

Caractéristique commune

Si on fait abstraction:

1. des intuitions géometriques
2. de la dynamique particulière de chaque algorithme, alors:

Boosting et SVM peuvent s'envisager comme des:

procédures de minimisation d'un risque convexe pénalisé dans des espaces fonctionnels massifs.

Elles sont caractérisées par

- des classes d'estimateurs différentes,
- des risques différents,

Caractéristique commune

Si on fait abstraction:

1. des intuitions géometriques
2. de la dynamique particulière de chaque algorithme, alors:

Boosting et SVM peuvent s'envisager comme des:

procédures de minimisation d'un risque convexe pénalisé dans des espaces fonctionnels massifs.

Elles sont caractérisées par

- des classes d'estimateurs différentes,
- des risques différents,
- des pénalités différentes.

Classe $\mathcal{F}$ d'estimateurs

- **Support Vector Machines** - soit k un noyau positif

$$\mathcal{F} = \left\{ f = \sum_{i=1}^{+\infty} \alpha_i k(x_i, \cdot) \ : \ \alpha_i \in \mathbb{R}, x_i \in \mathcal{X} \right\}$$

Classe $\mathcal{F}$ d'estimateurs

- **Support Vector Machines** - soit k un noyau positif

$$\mathcal{F} = \left\{ f = \sum_{i=1}^{+\infty} \alpha_i k(x_i, \cdot) \ : \ \alpha_i \in \mathbb{R}, x_i \in \mathcal{X} \right\}$$

- **Boosting** - soit $\mathcal{G}$ famille de classifieurs simples de VC dimension V finie

$$\mathcal{F} = \left\{ f = \sum_{i=1}^{+\infty} w_i g_i \ : \ w_i \in \mathbb{R}, g_i \in \mathcal{G} \right\}$$

Classe $\mathcal{F}$ d'estimateurs

- **Support Vector Machines** - soit k un noyau positif

$$\mathcal{F} = \left\{ f = \sum_{i=1}^{+\infty} \alpha_i k(x_i, \cdot) \ : \ \alpha_i \in \mathbb{R}, x_i \in \mathcal{X} \right\}$$

- **Boosting** - soit $\mathcal{G}$ famille de classifieurs simples de VC dimension V finie

$$\mathcal{F} = \left\{ f = \sum_{i=1}^{+\infty} w_i g_i \ : \ w_i \in \mathbb{R}, g_i \in \mathcal{G} \right\}$$

Classe $\mathcal{F}$ d'estimateurs

- **Support Vector Machines** - soit k un noyau positif

$$\mathcal{F} = \left\{ f = \sum_{i=1}^{+\infty} \alpha_i k(x_i, \cdot) \ : \ \alpha_i \in \mathbb{R}, x_i \in \mathcal{X} \right\}$$

- **Boosting** - soit $\mathcal{G}$ famille de classifieurs simples de VC dimension V finie

$$\mathcal{F} = \left\{ f = \sum_{i=1}^{+\infty} w_i g_i \ : \ w_i \in \mathbb{R}, g_i \in \mathcal{G} \right\}$$

Les deux algorithmes construisent des combinaisons linéaires de fonctions.

Cas des SVM - Retour sur la formulation lagrangienne I

On pose

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x)$$

$$\|f\|_{\mathcal{F}} = \sqrt{\sum_{i,j} \alpha_i \alpha_j k(x_i, x_j)}$$

Cas des SVM - Retour sur la formulation lagrangienne I

On pose

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x)$$

$$\|f\|_{\mathcal{F}} = \sqrt{\sum_{i,j} \alpha_i \alpha_j k(x_i, x_j)}$$

Formulation lagrangienne I

$$\min_{f \in \mathcal{F}} \frac{1}{2} \|f\|_{\mathcal{F}}^2 + C \sum_{i=1}^{n} \xi_i$$

sous les contraintes:

$$\forall i = 1, \ldots, n \ , \quad \xi_i \geq (1 - y_i \cdot f(x_i))_+$$

Interprétation - "hinge loss"

On pose : $\varphi(x) = (1+x)_+$ "hinge loss"

Interprétation - "hinge loss"

On pose : $\varphi(x) = (1+x)_+$ "hinge loss"

Minimisation d'un risque pénalisé

$$\min_{f \in \mathcal{F}} \frac{1}{2}\|f\|_{\mathcal{F}}^2 + C \sum_{i=1}^{n} \varphi(-y_i f(x_i))$$

Interprétation - "hinge loss"

On pose : $\varphi(x) = (1+x)_+$ "hinge loss"

Minimisation d'un risque pénalisé

$$\min_{f \in \mathcal{F}} \frac{1}{2}\|f\|_{\mathcal{F}}^2 + C\sum_{i=1}^{n} \varphi(-y_i f(x_i))$$

Commentaires:

- formulation importante pour la théorie statistique

Interprétation - "hinge loss"

On pose : $\varphi(x) = (1+x)_+$ "hinge loss"

Minimisation d'un risque pénalisé

$$\min_{f \in \mathcal{F}} \frac{1}{2}\|f\|_{\mathcal{F}}^2 + C \sum_{i=1}^{n} \varphi(-y_i f(x_i))$$

Commentaires:

- formulation importante pour la théorie statistique
- moins commode pour l'optimisation que la formulation duale

Interprétation - "hinge loss"

On pose : $\varphi(x) = (1+x)_+$ "hinge loss"

Minimisation d'un risque pénalisé

$$\min_{f \in \mathcal{F}} \frac{1}{2}\|f\|_{\mathcal{F}}^2 + C \sum_{i=1}^{n} \varphi(-y_i f(x_i))$$

Commentaires:

- formulation importante pour la théorie statistique
- moins commode pour l'optimisation que la formulation duale
- $\|f\|_{\mathcal{F}}$ fournit une mesure de régularité de f

Cadre général

- **Fonction de décision :**

$$f : \mathcal{X} \to \mathbb{R}$$

Cadre général

- **Fonction de décision :**

$$f \ : \ \mathcal{X} \to \mathbb{R}$$

- **Classifieur :**

$$g(x) = g_f(x) = \mathrm{sgn}(f(x)) \ \in \{-1, +1\}$$

Cadre général

- **Fonction de décision :**

$$f \ : \ \mathcal{X} \to \mathbb{R}$$

- **Classifieur :**

$$g(x) = g_f(x) = \operatorname{sgn}(f(x)) \ \in \{-1, +1\}$$

- **Critère naturel :**

$$L(f) = \mathbb{P}\{Y \cdot f(X) < 0\} = \mathbb{E}\left\{\mathbb{I}_{[Y \cdot f(X) < 0]}\right\}$$

φ-risque

- **Fonction de perte :** φ convexe, positive, telle que $\varphi(x) \geq \mathbb{I}_{\mathbb{R}_+}(x)$

φ-risque

- **Fonction de perte :** φ convexe, positive, telle que $\varphi(x) \geq \mathbb{I}_{\mathbb{R}_+}(x)$
- **Critère pratique (φ-risque) :**

$$\begin{aligned} A(f) &= \mathbb{E}\varphi(-Yf(X)) \\ &= \mathbb{E}\left[\eta(X)\varphi(-f(X)) + (1-\eta(X))\varphi(f(X))\right] \end{aligned}$$

où $\eta(X) = \mathbb{P}\{Y = 1 | X = x\}$

φ-risque

- **Fonction de perte :** φ convexe, positive, telle que $\varphi(x) \geq \mathbb{I}_{\mathbb{R}_+}(x)$
- **Critère pratique (φ-risque) :**

$$\begin{aligned} A(f) &= \mathbb{E}\varphi(-Yf(X)) \\ &= \mathbb{E}\left[\eta(X)\varphi(-f(X)) + (1-\eta(X))\varphi(f(X))\right] \end{aligned}$$

où $\eta(X) = \mathbb{P}\{Y = 1 | X = x\}$

φ-risque

- **Fonction de perte :** φ convexe, positive, telle que $\varphi(x) \geq \mathbb{I}_{\mathbb{R}_+}(x)$
- **Critère pratique (φ-risque) :**

$$\begin{aligned} A(f) &= \mathbb{E}\varphi(-Yf(X)) \\ &= \mathbb{E}\left[\eta(X)\varphi(-f(X)) + (1-\eta(X))\varphi(f(X))\right] \end{aligned}$$

où $\eta(X) = \mathbb{P}\{Y = 1 | X = x\}$

Question : minimiser A revient-il à minimiser L ?

φ-risque

- **Fonction de perte :** φ convexe, positive, telle que $\varphi(x) \geq \mathbb{I}_{\mathbb{R}_+}(x)$
- **Critère pratique (φ-risque) :**

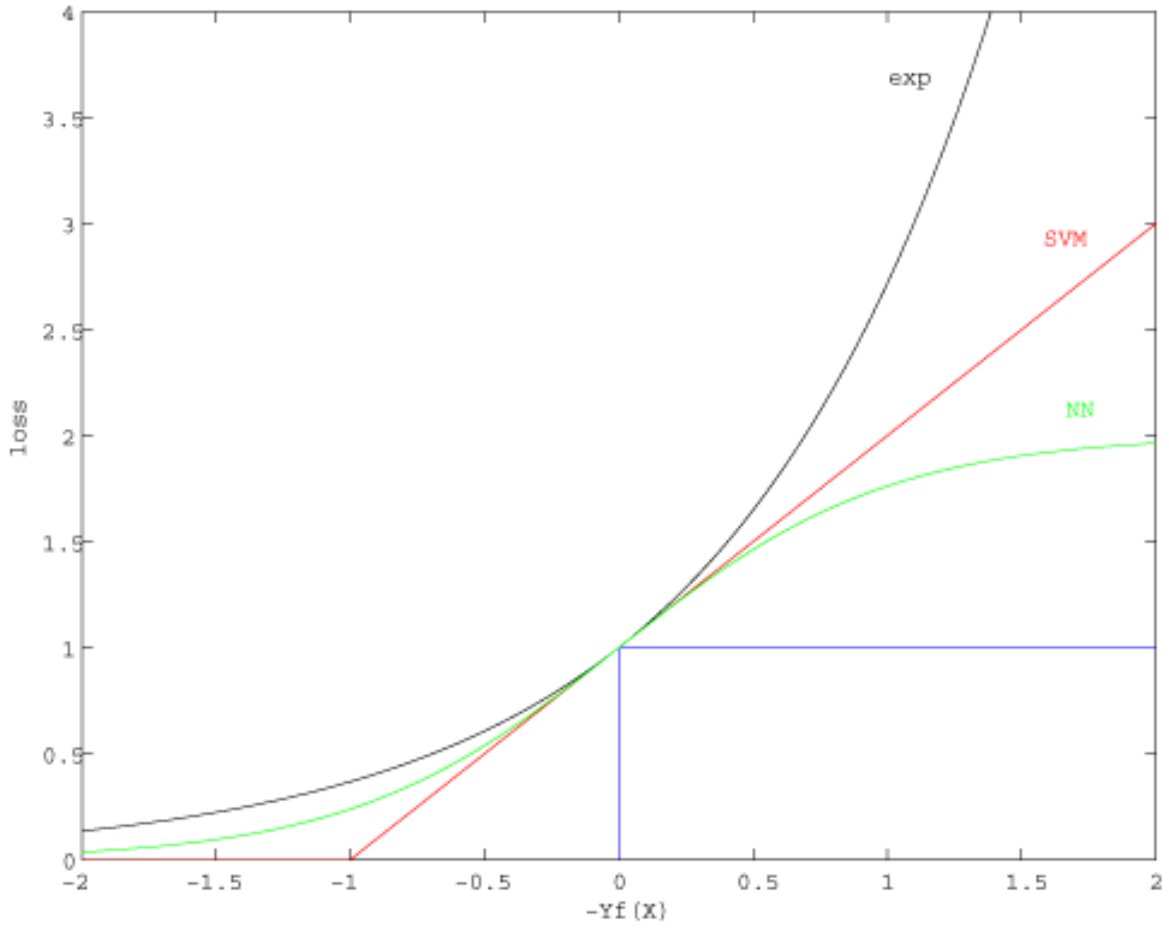
$$\begin{aligned} A(f) &= \mathbb{E}\varphi(-Yf(X)) \\ &= \mathbb{E}\left[\eta(X)\varphi(-f(X)) + (1-\eta(X))\varphi(f(X))\right] \end{aligned}$$

où $\eta(X) = \mathbb{P}\{Y = 1 | X = x\}$

Question : minimiser A revient-il à minimiser L ?

On a seulement que: $L(f) \leq A(f)$...

Fonctions de perte



Fonctions cibles

- Minimiseur de la fonctionnelle A noté f^*

Fonctions cibles

- Minimiseur de la fonctionnelle A noté f^*
- Minimum du φ-risque : $A^* = \min_f A(f) = A(f^*)$

Fonctions cibles

- Minimiseur de la fonctionnelle A noté f^*
- Minimum du φ-risque : $A^* = \min_f A(f) = A(f^*)$
- On peut montrer que $\operatorname{sgn}(f^*) = g^*$ le classifieur optimal

Fonctions cibles

- Minimiseur de la fonctionnelle A noté f^*
- Minimum du φ-risque : $A^* = \min_f A(f) = A(f^*)$
- On peut montrer que $\operatorname{sgn}(f^*) = g^*$ le classifieur optimal
- On peut également montrer que l'excès de risque en erreur de classification $L(f) - L^*$ est contrôlé par $A(f) - A^*$

Fonctions cibles

- Minimiseur de la fonctionnelle A noté f^*
- Minimum du φ-risque : $A^* = \min_f A(f) = A(f^*)$
- On peut montrer que $\operatorname{sgn}(f^*) = g^*$ le classifieur optimal
- On peut également montrer que l'excès de risque en erreur de classification $L(f) - L^*$ est contrôlé par $A(f) - A^*$
- Exemples:

Fonctions cibles

- Minimiseur de la fonctionnelle A noté f^*
- Minimum du φ-risque : $A^* = \min_f A(f) = A(f^*)$
- On peut montrer que $\operatorname{sgn}(f^*) = g^*$ le classifieur optimal
- On peut également montrer que l'excès de risque en erreur de classification $L(f) - L^*$ est contrôlé par $A(f) - A^*$
- Exemples:
  - perte exponentielle (boosting)

$$f^*(x) = \frac{1}{2} \log \left(\frac{\eta(x)}{1 - \eta(x)} \right)$$

Fonctions cibles

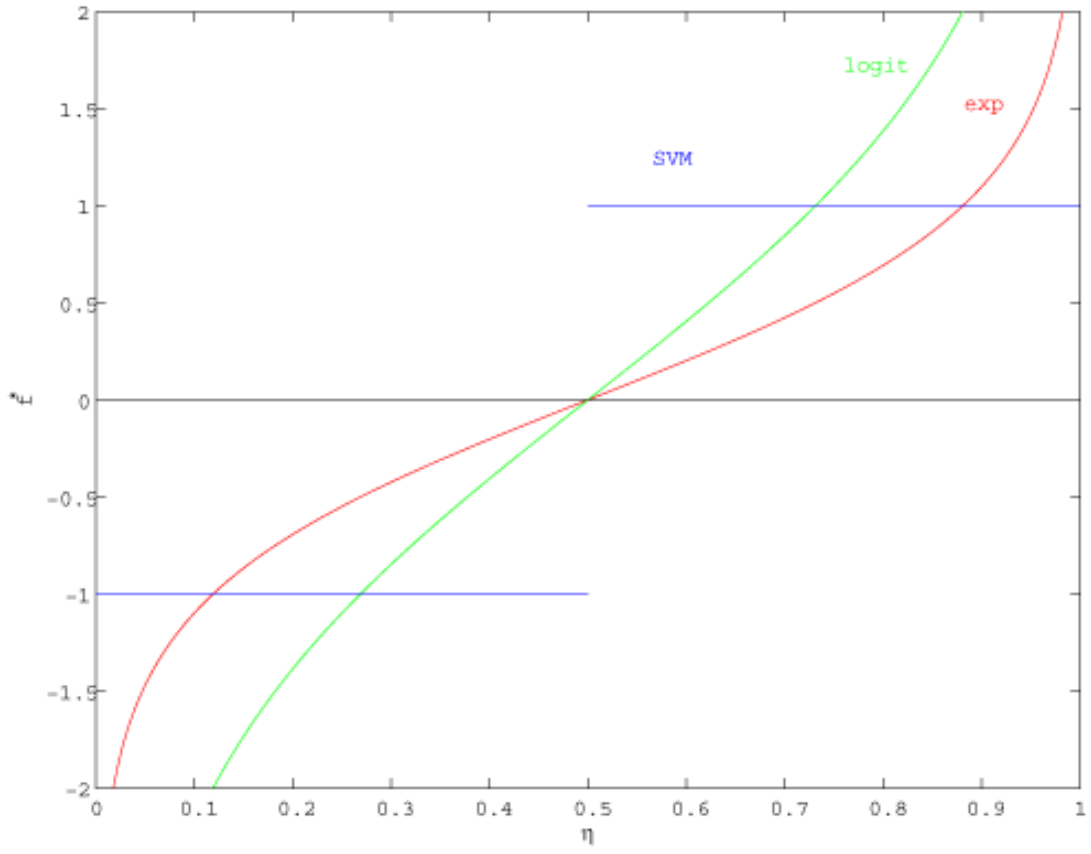
- Minimiseur de la fonctionnelle A noté f^*
- Minimum du φ-risque : $A^* = \min_f A(f) = A(f^*)$
- On peut montrer que $\operatorname{sgn}(f^*) = g^*$ le classifieur optimal
- On peut également montrer que l'excès de risque en erreur de classification $L(f) - L^*$ est contrôlé par $A(f) - A^*$
- Exemples:
  - perte exponentielle (boosting)

$$f^*(x) = \frac{1}{2} \log \left(\frac{\eta(x)}{1 - \eta(x)} \right)$$

  - "hinge loss" (SVM)

$$f^*(x) = \operatorname{sgn}(\eta(x) - 1/2) = g^*(x)$$

Fonctions cibles



Et après?

Jusqu' à présent, on a mis en évidence :

1. les fonctions de décision candidates sont des combinaisons linéaires de fonctions

Et après?

Jusqu' à présent, on a mis en évidence :

1. les fonctions de décision candidates sont des combinaisons linéaires de fonctions
2. les critères de risque sont convexifiés

Et après?

Jusqu' à présent, on a mis en évidence :

1. les fonctions de décision candidates sont des combinaisons linéaires de fonctions
2. les critères de risque sont convexifiés

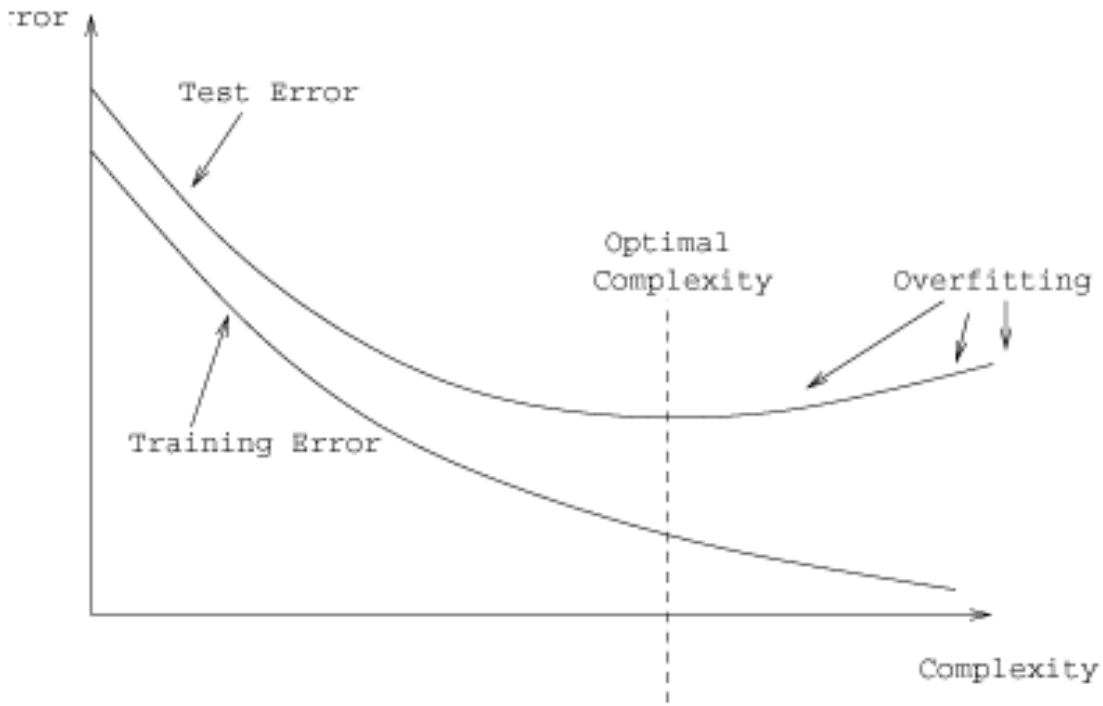
Et après?

Jusqu' à présent, on a mis en évidence :

1. les fonctions de décision candidates sont des combinaisons linéaires de fonctions
2. les critères de risque sont convexifiés

Quelle théorie pour le contrôle de la complexité ?

Overfitting



Théorie de la complexité de Vapnik-Chervonenkis

- Notion de VC dimension

Théorie de la complexité de Vapnik-Chervonenkis

- Notion de VC dimension
  - Mesure de complexité de nature combinatoire

Théorie de la complexité de Vapnik-Chervonenkis

- Notion de VC dimension
  - Mesure de complexité de nature combinatoire
  - Capacité d'une famille d'ensembles à séparer un jeu de points

Théorie de la complexité de Vapnik-Chervonenkis

- Notion de VC dimension
  - Mesure de complexité de nature combinatoire
  - Capacité d'une famille d'ensembles à séparer un jeu de points
- Exemple : $\mathcal{H}$ famille de demi-espaces sur $\mathbb{R}^m$, $\text{VCdim}(\mathcal{H}) = m + 1$

Théorie de la complexité de Vapnik-Chervonenkis

- Notion de VC dimension
  - Mesure de complexité de nature combinatoire
  - Capacité d'une famille d'ensembles à séparer un jeu de points
- Exemple : $\mathcal{H}$ famille de demi-espaces sur $\mathbb{R}^m$, $\text{VCdim}(\mathcal{H}) = m + 1$
- Problème pour interpréter les performances des SVM ($m = \infty$)

Théorie de la complexité de Vapnik-Chervonenkis

- Notion de VC dimension
  - Mesure de complexité de nature combinatoire
  - Capacité d'une famille d'ensembles à séparer un jeu de points
- Exemple : $\mathcal{H}$ famille de demi-espaces sur $\mathbb{R}^m$, $\text{VCdim}(\mathcal{H}) = m + 1$
- Problème pour interpréter les performances des SVM ($m = \infty$)
- Idem avec le boosting...

Théorie de la complexité de Vapnik-Chervonenkis

- Notion de VC dimension
  - Mesure de complexité de nature combinatoire
  - Capacité d'une famille d'ensembles à séparer un jeu de points
- Exemple : $\mathcal{H}$ famille de demi-espaces sur $\mathbb{R}^m$, $\text{VCdim}(\mathcal{H}) = m + 1$
- Problème pour interpréter les performances des SVM ($m = \infty$)
- Idem avec le boosting...

Théorie de la complexité de Vapnik-Chervonenkis

- Notion de VC dimension
  - Mesure de complexité de nature combinatoire
  - Capacité d'une famille d'ensembles à séparer un jeu de points
- Exemple : $\mathcal{H}$ famille de demi-espaces sur $\mathbb{R}^m$, $\text{VCdim}(\mathcal{H}) = m + 1$
- Problème pour interpréter les performances des SVM ($m = \infty$)
- Idem avec le boosting...

Les algorithmes performants ne sont pas rigoureusement expliqués par la théorie "standard" de Vapnik-Chervonenkis.

Autre mesure de complexité

- $\mathcal{F}$ une famille de fonctions de décision et D_n un jeu de points $X_1, \ldots, X_n$

Autre mesure de complexité

- $\mathcal{F}$ une famille de fonctions de décision et D_n un jeu de points $X_1, \ldots, X_n$

Complexité de Rademacher

Soient $\epsilon_1, \ldots, \epsilon_n$ variables de signes +1/-1 i.i.d.

$$R_n(\mathcal{F}) = \mathbb{E}_\epsilon \sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i f(X_i) \right|$$

Autre mesure de complexité

- $\mathcal{F}$ une famille de fonctions de décision et D_n un jeu de points $X_1, \ldots, X_n$

Complexité de Rademacher

Soient $\epsilon_1, \ldots, \epsilon_n$ variables de signes +1/-1 i.i.d.

$$R_n(\mathcal{F}) = \mathbb{E}_\epsilon \sup_{f \in \mathcal{F}} \left| \frac{1}{n} \sum_{i=1}^{n} \epsilon_i f(X_i) \right|$$

Cas des familles $\mathcal{F} = \{f(x) = \sum \alpha_i k(x_i, x) \ : \ \sum_{i,j} \alpha_i \alpha_j k(x_i, x_j) \leq B^2\}$

Résultat:

$$R_n(\mathcal{F}) \leq \frac{B}{n} \sqrt{\sum_{i=1}^{n} k(X_i, X_i)}$$

Exemples II - données réelles

Validation croisée

- **Problème :** estimation de l'erreur en généralisation $\mathbb{P}\{Y \cdot f(X) < 0\}$

Validation croisée

- **Problème :** estimation de l'erreur en généralisation $\mathbb{P}\{Y \cdot f(X) < 0\}$
- **Une solution :** on réserve une partie des données initiales pour constituer une base de test obtenue par sous-échantillonnage aléatoirement

Validation croisée

- **Problème :** estimation de l'erreur en généralisation $\mathbb{P}\{Y \cdot f(X) < 0\}$
- **Une solution :** on réserve une partie des données initiales pour constituer une base de test obtenue par sous-échantillonnage aléatoirement
- **Une autre approche :** la validation croisée

Validation croisée

- **Problème :** estimation de l'erreur en généralisation $\mathbb{P}\{Y \cdot f(X) < 0\}$
- **Une solution :** on réserve une partie des données initiales pour constituer une base de test obtenue par sous-échantillonnage aléatoirement
- **Une autre approche :** la validation croisée
  - on coupe l'échantillon D (taille n) en p sous-échantillons $D_1, \ldots, D_p$ de taille n/p

Validation croisée

- **Problème :** estimation de l'erreur en généralisation $\mathbb{P}\{Y \cdot f(X) < 0\}$
- **Une solution :** on réserve une partie des données initiales pour constituer une base de test obtenue par sous-échantillonnage aléatoirement
- **Une autre approche :** la validation croisée
  - on coupe l'échantillon D (taille n) en p sous-échantillons $D_1, \ldots, D_p$ de taille n/p
  - on apprend sur $D \setminus D_i$ de taille $n - n/p$

Validation croisée

- **Problème :** estimation de l'erreur en généralisation $\mathbb{P}\{Y \cdot f(X) < 0\}$
- **Une solution :** on réserve une partie des données initiales pour constituer une base de test obtenue par sous-échantillonnage aléatoirement
- **Une autre approche :** la validation croisée
  - on coupe l'échantillon D (taille n) en p sous-échantillons $D_1, \ldots, D_p$ de taille n/p
  - on apprend sur $D \setminus D_i$ de taille $n - n/p$
  - on teste sur D_i

Validation croisée

- **Problème :** estimation de l'erreur en généralisation $\mathbb{P}\{Y \cdot f(X) < 0\}$
- **Une solution :** on réserve une partie des données initiales pour constituer une base de test obtenue par sous-échantillonnage aléatoirement
- **Une autre approche :** la validation croisée
  - on coupe l'échantillon D (taille n) en p sous-échantillons $D_1, \ldots, D_p$ de taille n/p
  - on apprend sur $D \setminus D_i$ de taille $n - n/p$
  - on teste sur D_i
  - on fait la moyenne des erreurs

Validation croisée

- **Problème :** estimation de l'erreur en généralisation $\mathbb{P}\{Y \cdot f(X) < 0\}$
- **Une solution :** on réserve une partie des données initiales pour constituer une base de test obtenue par sous-échantillonnage aléatoirement
- **Une autre approche :** la validation croisée
  - on coupe l'échantillon D (taille n) en p sous-échantillons $D_1, \ldots, D_p$ de taille n/p
  - on apprend sur $D \setminus D_i$ de taille $n - n/p$
  - on teste sur D_i
  - on fait la moyenne des erreurs
  - en général $p = 10$

Erreur d'apprentissage vs. erreur de test

Script :

```
load spam.data
N = size(spam,1); ncol = size(spam, 2); d = ncol-1;
m = floor(N*.1); n = N-m;
ind = randperm(N);
X = spam(ind, 1:d);
Y = 2*spam(ind,ncol)-1;
d1 = data(X(1:n, :), Y(1:n));
d2 = data(X((n+1):N, :), Y((n+1):N));
a=svm;
[tr a] = train(a,d1);
r = test(a, d2);
loss(tr)
loss(r)
```

Utiliser la validation croisée

Il faut remplacer:

```
a=svm;
```

par

```
s=svm;
a=cv(s, 'folds=10');
```

On récupère l'erreur correspondante avec :

```
tmp = tmp.Y
```

Noyaux positifs

Définition

Soit $\mathcal{X}$ l'espace où vivent les observations.

Noyau positif

Une fonction $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ est un noyau positif si et seulement si

1. k est symétrique: $k(x, x') = k(x', x)$, $\forall x, x' \in \mathcal{X}$
2. k est positive:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(x_i, x_j) \geq 0, \quad \forall c_i \in \mathbb{R}, \quad \forall x_i \in \mathcal{X}, \quad \forall n \geq 1$$

Un théorème d'analyse

Théorème de Mercer

Pour tout noyau positif k sur $\mathcal{X}$ il existe un espace de Hilbert $\mathcal{H}$ et une application Φ tels que:

$$k(x, x') = < \Phi(x), \Phi(x') >, \quad \forall x, x' \in \mathcal{X}$$

où $< , >$ représente le produit scalaire sur $\mathcal{H}$.

Commentaires

- Le théorème de Mercer est non constructif: il ne fournit ni $\mathcal{H}$, ni Φ

Commentaires

- Le théorème de Mercer est non constructif: il ne fournit ni $\mathcal{H}$, ni Φ
- En pratique:

Commentaires

- Le théorème de Mercer est non constructif: il ne fournit ni $\mathcal{H}$, ni Φ
- En pratique:
  - $\mathcal{H}$ est un espace de grande dimension

Commentaires

- Le théorème de Mercer est non constructif: il ne fournit ni $\mathcal{H}$, ni Φ
- En pratique:
  - $\mathcal{H}$ est un espace de grande dimension
  - Φ est une application non-linéaire

Commentaires

- Le théorème de Mercer est non constructif: il ne fournit ni $\mathcal{H}$, ni Φ
- En pratique:
  - $\mathcal{H}$ est un espace de grande dimension
  - Φ est une application non-linéaire
- $\mathcal{H}$ est un espace de représentation des données, connu sous le nom de "feature space" ou espace de caractéristiques

Commentaires

- Le théorème de Mercer est non constructif: il ne fournit ni $\mathcal{H}$, ni Φ
- En pratique:
  - $\mathcal{H}$ est un espace de grande dimension
  - Φ est une application non-linéaire
- $\mathcal{H}$ est un espace de représentation des données, connu sous le nom de "feature space" ou espace de caractéristiques
- L'astuce du noyau consiste à faire l'impasse sur $\mathcal{H}$ et Φ si on sait qu'ils existent!

Distances images

- Norme euclidienne sur $\mathbb{R}^m$: $\forall u \in \mathbb{R}^m,\ \|u\| = \sqrt{< u, u >}$ où $< , >$ produit scalaire sur $\mathbb{R}^m$

Distances images

- Norme euclidienne sur $\mathbb{R}^m$: $\forall u \in \mathbb{R}^m,\ \|u\| = \sqrt{< u, u >}$ où $< , >$ produit scalaire sur $\mathbb{R}^m$
- Distance euclidienne:
  $d(u, v) = \|u - v\| = \sqrt{< u, u > + < v, v > -2 < u, v >}$

Distances images

- Norme euclidienne sur $\mathbb{R}^m$: $\forall u \in \mathbb{R}^m,\ \|u\| = \sqrt{< u, u >}$ où $< , >$ produit scalaire sur $\mathbb{R}^m$
- Distance euclidienne:
  $d(u, v) = \|u - v\| = \sqrt{< u, u > + < v, v > -2 < u, v >}$
- Transformation non-linéaire : $\Phi\ :\ \mathbb{R}^d \rightarrow \mathbb{R}^m$ avec $m > d$

Distances images

- Norme euclidienne sur $\mathbb{R}^m$: $\forall u \in \mathbb{R}^m,\ \|u\| = \sqrt{< u, u >}$ où $< , >$ produit scalaire sur $\mathbb{R}^m$
- Distance euclidienne:
  $d(u, v) = \|u - v\| = \sqrt{< u, u > + < v, v > -2 < u, v >}$
- Transformation non-linéaire : $\Phi\ :\ \mathbb{R}^d \rightarrow \mathbb{R}^m$ avec $m > d$
- Noyau: $k(x, x') =< \Phi(x), \Phi(x') >$

Distances images

- Norme euclidienne sur $\mathbb{R}^m$: $\forall u \in \mathbb{R}^m,\ \|u\| = \sqrt{< u, u >}$ où $< , >$ produit scalaire sur $\mathbb{R}^m$
- Distance euclidienne:
  $d(u, v) = \|u - v\| = \sqrt{< u, u > + < v, v > -2 < u, v >}$
- Transformation non-linéaire : $\Phi\ :\ \mathbb{R}^d \rightarrow \mathbb{R}^m$ avec $m > d$
- Noyau: $k(x, x') =< \Phi(x), \Phi(x') >$
- Distance image:

$$d_\Phi(x, x') = \|\Phi(x) - \Phi(x')\| = \sqrt{k(x, x) + k(x', x') - 2k(x, x')}$$

$\Rightarrow$ la distance induite par Φ ne fait intervenir que le noyau

Intérêt pour la classification

- Aucune complication algorithmique en remplaçant le produit scalaire par une autre mesure de similarité

Intérêt pour la classification

- Aucune complication algorithmique en remplaçant le produit scalaire par une autre mesure de similarité
- Transformer un problème initialement non-linéaire en un problème linéaire en envoyant les données dans un espace plus grand

Intérêt pour la classification

- Aucune complication algorithmique en remplaçant le produit scalaire par une autre mesure de similarité
- Transformer un problème initialement non-linéaire en un problème linéaire en envoyant les données dans un espace plus grand

Intérêt pour la classification

- Aucune complication algorithmique en remplaçant le produit scalaire par une autre mesure de similarité
- Transformer un problème initialement non-linéaire en un problème linéaire en envoyant les données dans un espace plus grand

Exemple

Soit $f(x, y) = ax^2 + bx + c - y = 0$ une surface de décision polynomiale (parabole dans $\mathbb{R}^2$).

Intérêt pour la classification

- Aucune complication algorithmique en remplaçant le produit scalaire par une autre mesure de similarité
- Transformer un problème initialement non-linéaire en un problème linéaire en envoyant les données dans un espace plus grand

Exemple

Soit $f(x, y) = ax^2 + bx + c - y = 0$ une surface de décision polynomiale (parabole dans $\mathbb{R}^2$).

Rôle clé de la transformation:

$$\begin{array}{rccc} \Phi & : & \mathbb{R}^2 & \rightarrow & \mathbb{R}^4 \\ & & x & \mapsto & \left(x^2, x, 1, y\right)^T \end{array}$$

Du non-linéaire au linéaire

Exemple (suite)

On peut écrire:

$$g(x^2, x, 1, y) = ax^2 + bx + c - y = 0$$

où $g(u, v, w, y) = au + bv + cw - y$.

L'équation $g(u, v, w, y) = 0$ définit une surface de décision linéaire dans $\mathbb{R}^4$.

Du non-linéaire au linéaire

Exemple (suite)

On peut écrire:

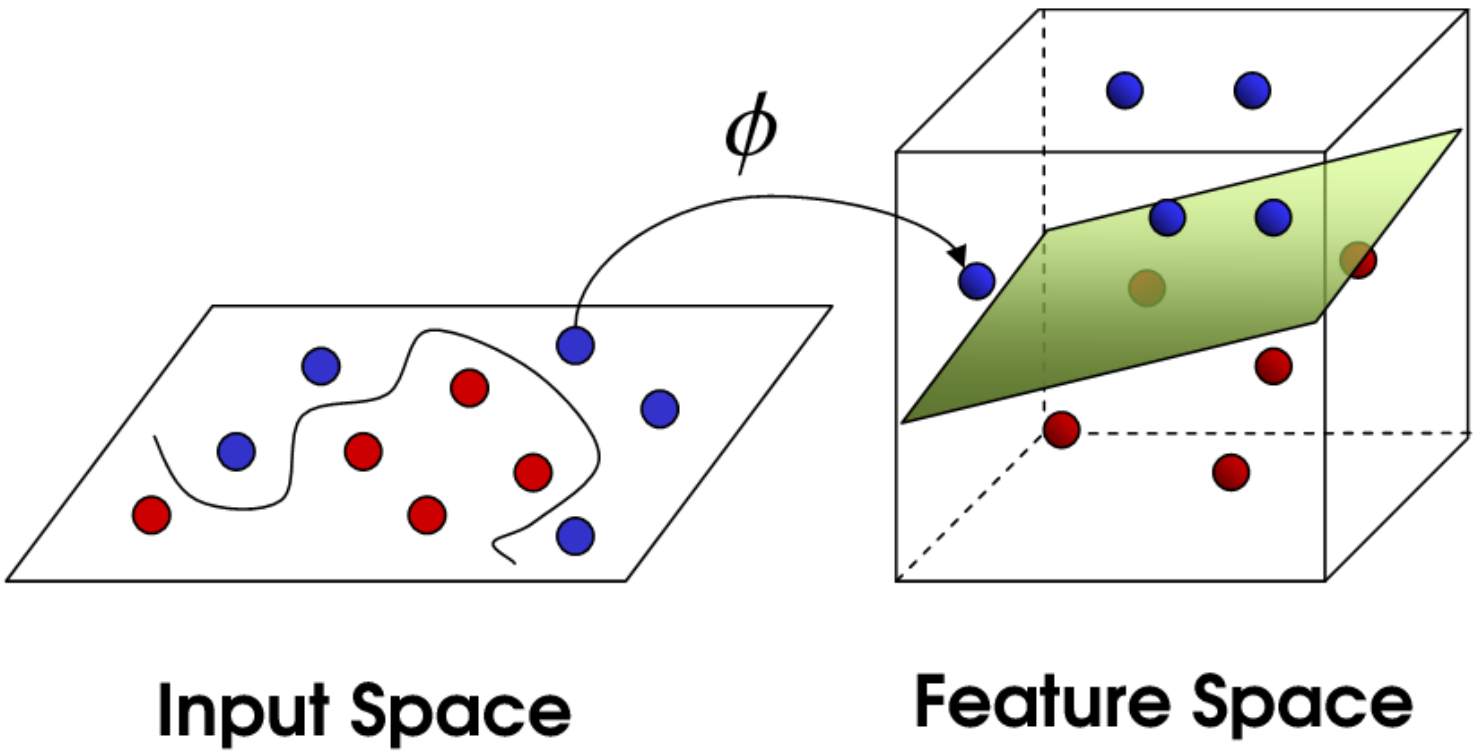
$$g(x^2, x, 1, y) = ax^2 + bx + c - y = 0$$

où $g(u, v, w, y) = au + bv + cw - y$.

L'équation $g(u, v, w, y) = 0$ définit une surface de décision linéaire dans $\mathbb{R}^4$.

Un problème non-linéaire dans un certain espace peut parfois se formuler comme un problème linéaire dans un espace plus grand.

Du non-linéaire au linéaire



ACP à noyau

ACP classique

On considère un nuage de points $x_1, \ldots, x_n$ centrés en l'origine.

ACP classique

On considère un nuage de points $x_1, \ldots, x_n$ centrés en l'origine.

Buts de l'ACP

- méthode de visualisation des données
- réduction de la dimension effective des données

ACP classique

On considère un nuage de points $x_1, \ldots, x_n$ centrés en l'origine.

Buts de l'ACP

- méthode de visualisation des données
- réduction de la dimension effective des données

L'ACP consiste à identifier les composantes principales de l'échantillon constituées par

1. la meilleure direction de projection du nuage de points
   i.e. celle de variance maximale

ACP classique

On considère un nuage de points $x_1, \ldots, x_n$ centrés en l'origine.

Buts de l'ACP

- méthode de visualisation des données
- réduction de la dimension effective des données

L'ACP consiste à identifier les composantes principales de l'échantillon constituées par

1. la meilleure direction de projection du nuage de points
   i.e. celle de variance maximale
2. puis, la meilleure direction de projection orthogonale à la première

ACP classique

On considère un nuage de points $x_1, \ldots, x_n$ centrés en l'origine.

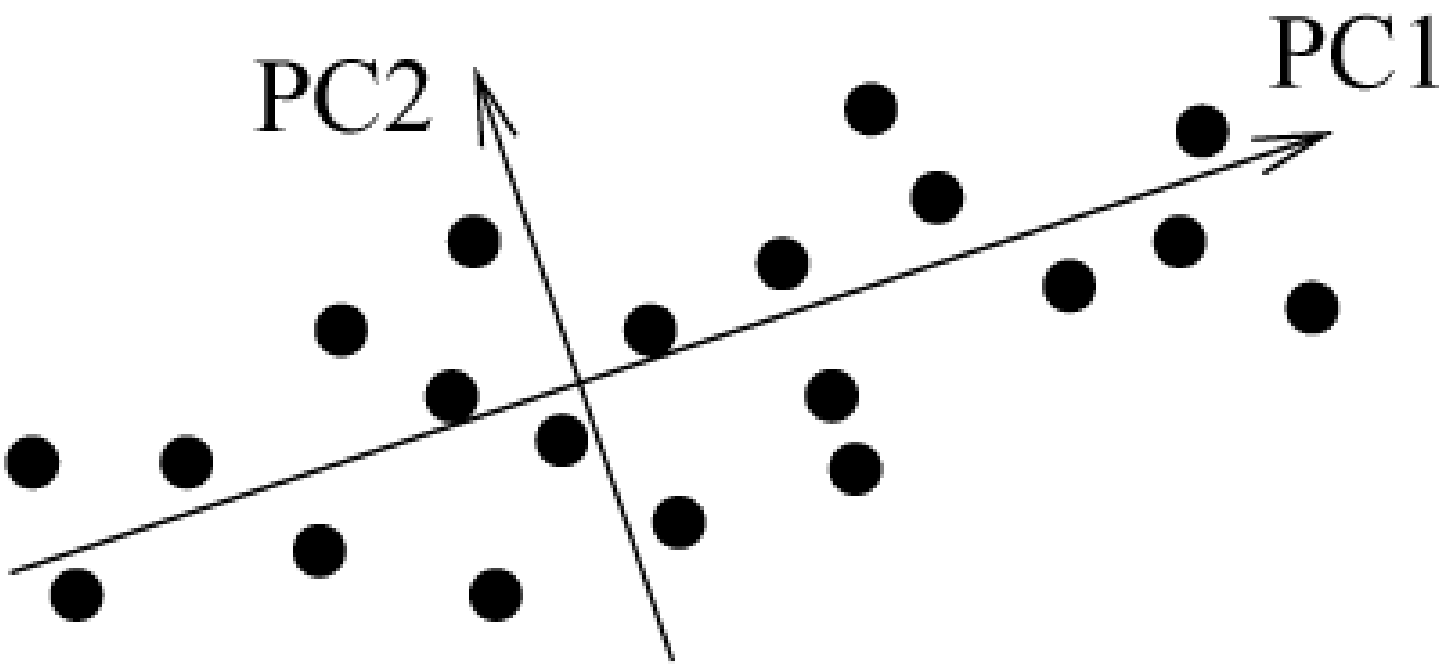
Buts de l'ACP

- méthode de visualisation des données
- réduction de la dimension effective des données

L'ACP consiste à identifier les composantes principales de l'échantillon constituées par

1. la meilleure direction de projection du nuage de points
   i.e. celle de variance maximale
2. puis, la meilleure direction de projection orthogonale à la première
3. et, ainsi de suite, jusqu'à la n-ième

ACP



ACP (suite)

- Projection orthogonale d'un vecteur x sur la direction $w \in \mathbb{R}^d$:

$$p_w(x) = \frac{< x, w >}{\|w\|}$$

ACP (suite)

- Projection orthogonale d'un vecteur x sur la direction $w \in \mathbb{R}^d$:

$$p_w(x) = \frac{< x, w >}{\|w\|}$$

- Variance empirique du nuage de points selon la direction w:

$$\mathbb{V}(p_w) = \frac{1}{n} \sum_{i=1}^{n} \frac{< x_i, w >^2}{\|w\|^2}$$

ACP (suite)

- Projection orthogonale d'un vecteur x sur la direction $w \in \mathbb{R}^d$:

$$p_w(x) = \frac{< x, w >}{\|w\|}$$

- Variance empirique du nuage de points selon la direction w:

$$\mathbb{V}(p_w) = \frac{1}{n} \sum_{i=1}^{n} \frac{< x_i, w >^2}{\|w\|^2}$$

- Matrice de covariance empirique $\Sigma = \frac{1}{n} \sum_{i=1}^{n} x_i \; x_i^T$

ACP (suite)

- Projection orthogonale d'un vecteur x sur la direction $w \in \mathbb{R}^d$:

$$p_w(x) = \frac{< x, w >}{\|w\|}$$

- Variance empirique du nuage de points selon la direction w:

$$\mathbb{V}(p_w) = \frac{1}{n} \sum_{i=1}^{n} \frac{< x_i, w >^2}{\|w\|^2}$$

- Matrice de covariance empirique $\Sigma = \frac{1}{n} \sum_{i=1}^{n} x_i \; x_i^T$
- On a donc :

$$\mathbb{V}(p_w) = \frac{w^T \Sigma w}{\|w\|^2}$$

Problème d'optimisation

Première composante principale

$$\arg\max_{w} \mathbb{V}(p_w) = \frac{w^T \Sigma w}{\|w\|^2}$$

Problème d'optimisation

Première composante principale

$$\arg\max_{w} \mathbb{V}(p_w) = \frac{w^T \Sigma w}{\|w\|^2}$$

Solution

Les composantes principales sont les vecteurs propres de la Σ rangés selon la décroissance des valeurs propres correspondantes.

Problème d'optimisation

Première composante principale

$$\arg\max_{w} \mathbb{V}(p_w) = \frac{w^T \Sigma w}{\|w\|^2}$$

Solution

Les composantes principales sont les vecteurs propres de la Σ rangés selon la décroissance des valeurs propres correspondantes.

Remarque : la matrice Σ est symétrique réelle donc diagonalisable dans une base orthonormée.

ACP (suite)

On cherche un vecteur v et un réel λ tels que:

$$\Sigma v = \lambda v$$

Or, on a :

$$\Sigma v = \frac{1}{n} \sum_{i=1}^{n} < x_i, v > \ x_i$$

D'où:

$$v = \sum_{i=1}^{n} \left(\frac{< x_i, v >}{n\lambda} \right) x_i = \sum_{i=1}^{n} \alpha_i x_i$$

ACP (suite)

On cherche un vecteur v et un réel λ tels que:

$$\Sigma v = \lambda v$$

Or, on a :

$$\Sigma v = \frac{1}{n} \sum_{i=1}^{n} < x_i, v > \ x_i$$

D'où:

$$v = \sum_{i=1}^{n} \left(\frac{< x_i, v >}{n\lambda} \right) x_i = \sum_{i=1}^{n} \alpha_i x_i$$

On utilise

$$x_j^T \Sigma v = \lambda < x_j, v >, \quad \forall j$$

et on y substitue les expressions de Σ et v:

$$\frac{1}{n} \sum_{i=1}^{n} \alpha_i \left\langle x_j, \sum_{k=1}^{n} < x_k, x_i > \ x_k \right\rangle = \lambda \sum_{i=1}^{n} \alpha_i < x_j, x_i >$$

ACP (suite)

- On note $K = (< x_i, x_j >)_{i,j}$ la matrice de Gram

ACP (suite)

- On note $K = (< x_i, x_j >)_{i,j}$ la matrice de Gram
- On peut écrire alors le système:

$$K^2\alpha = n\lambda K\alpha$$

ACP (suite)

- On note $K = (< x_i, x_j >)_{i,j}$ la matrice de Gram
- On peut écrire alors le système:

$$K^2\alpha = n\lambda K\alpha$$

- Pour résoudre en α, on résout donc le problème aux éléments propres

$$K\alpha = n\lambda\alpha$$

Défauts de l'ACP classique

- elle est adaptée surtout au cas de réalisations de gaussiennes multivariées

Défauts de l'ACP classique

- elle est adaptée surtout au cas de réalisations de gaussiennes multivariées
  - en général, la non-corrélation n'implique pas l'indépendance des directions principales

Défauts de l'ACP classique

- elle est adaptée surtout au cas de réalisations de gaussiennes multivariées
  - en général, la non-corrélation n'implique pas l'indépendance des directions principales
  - alternative : Analyse en Composantes Indépendantes (plutôt que Principales)

Défauts de l'ACP classique

- elle est adaptée surtout au cas de réalisations de gaussiennes multivariées
  - en général, la non-corrélation n'implique pas l'indépendance des directions principales
  - alternative : Analyse en Composantes Indépendantes (plutôt que Principales)
- elle est adaptée aux structures linéaires

Défauts de l'ACP classique

- elle est adaptée surtout au cas de réalisations de gaussiennes multivariées
  - en général, la non-corrélation n'implique pas l'indépendance des directions principales
  - alternative : Analyse en Composantes Indépendantes (plutôt que Principales)
- elle est adaptée aux structures linéaires
  - les nuages de points ne sont pas tous ellipsoidaux!!

Défauts de l'ACP classique

- elle est adaptée surtout au cas de réalisations de gaussiennes multivariées
  - en général, la non-corrélation n'implique pas l'indépendance des directions principales
  - alternative : Analyse en Composantes Indépendantes (plutôt que Principales)
- elle est adaptée aux structures linéaires
  - les nuages de points ne sont pas tous ellipsoidaux!!
  - alternative : Kernel PCA

ACP à noyau

- On applique une transformation Φ qui envoie le nuage de points X dans un espace où la structure est linéaire

ACP à noyau

- On applique une transformation Φ qui envoie le nuage de points X dans un espace où la structure est linéaire
- La matrice de covariance de $\Phi(X) = (\Phi(x_1), \ldots, \Phi(x_n))^T$ est alors

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i) \Phi(x_i)^T$$

ACP à noyau

- On applique une transformation Φ qui envoie le nuage de points X dans un espace où la structure est linéaire
- La matrice de covariance de $\Phi(X) = (\Phi(x_1), \ldots, \Phi(x_n))^T$ est alors

$$\Sigma = \frac{1}{n} \sum_{i=1}^{n} \Phi(x_i)\Phi(x_i)^T$$

- **Astuce du noyau :** $K = \left(k(x_i, x_j)\right)_{i,j} = \left(\Phi(x_i)^T \Phi(x_j)\right)_{i,j}$

ACP à noyau (suite)

- "Directions" principales de la forme:

$$p_i(x) = \sum_{j=1}^{n} \alpha_{i,j} k(x_j, x)$$

ACP à noyau (suite)

- "Directions" principales de la forme:

$$p_i(x) = \sum_{j=1}^{n} \alpha_{i,j} k(x_j, x)$$

- le vecteur $\alpha_i = (\alpha_{i,1}, \ldots, \alpha_{i,n})$ est solution du problème d'optimisation:

$$\max_{\alpha} \frac{\alpha^T K^2 \alpha}{\alpha^T K \alpha}$$

sous les contraintes: $\alpha_i^T K \alpha_j$ pour $j = 1, \ldots, i-1$

ACP à noyau (suite)

- "Directions" principales de la forme:

$$p_i(x) = \sum_{j=1}^{n} \alpha_{i,j} k(x_j, x)$$

- le vecteur $\alpha_i = (\alpha_{i,1}, \ldots, \alpha_{i,n})$ est solution du problème d'optimisation:

$$\max_{\alpha} \frac{\alpha^T K^2 \alpha}{\alpha^T K \alpha}$$

  sous les contraintes: $\alpha_i^T K \alpha_j$ pour $j = 1, \ldots, i-1$

- on résout le problème aux éléments propres:

$$K\alpha = n\lambda\alpha$$

SVM pour la régression

Problème de la régression

- Problème de prédiction d'une variable d'intérêt réelle à partir d'un vecteur $X \in \mathbb{R}^d$ de variables explicatives

Problème de la régression

- Problème de prédiction d'une variable d'intérêt réelle à partir d'un vecteur $X \in \mathbb{R}^d$ de variables explicatives
- **Données :** $\{(x_i, y_i)\}_{i=1,\ldots,n}$

Problème de la régression

- Problème de prédiction d'une variable d'intérêt réelle à partir d'un vecteur $X \in \mathbb{R}^d$ de variables explicatives
- **Données :** $\{(x_i, y_i)\}_{i=1,\dots,n}$
- **Prédicteur :** $f(x) = \beta^T x + b$, $\quad \beta \in \mathbb{R}^d$, $b \in \mathbb{R}$

Problème de la régression

- Problème de prédiction d'une variable d'intérêt réelle à partir d'un vecteur $X \in \mathbb{R}^d$ de variables explicatives
- **Données :** $\{(x_i, y_i)\}_{i=1,\dots,n}$
- **Prédicteur :** $f(x) = \beta^T x + b$, $\beta \in \mathbb{R}^d$, $b \in \mathbb{R}$
- **Norme :** $\|f\| = \|\beta\| = \sqrt{\beta^T \beta}$

Problème de la régression

- Problème de prédiction d'une variable d'intérêt réelle à partir d'un vecteur $X \in \mathbb{R}^d$ de variables explicatives
- **Données :** $\{(x_i, y_i)\}_{i=1,\dots,n}$
- **Prédicteur :** $f(x) = \beta^T x + b$, $\beta \in \mathbb{R}^d$, $b \in \mathbb{R}$
- **Norme :** $\|f\| = \|\beta\| = \sqrt{\beta^T \beta}$
- **Critère :** moindres carrés $(y - f(x))^2$

Problème de la régression

- Problème de prédiction d'une variable d'intérêt réelle à partir d'un vecteur $X \in \mathbb{R}^d$ de variables explicatives
- **Données :** $\{(x_i, y_i)\}_{i=1,\dots,n}$
- **Prédicteur :** $f(x) = \beta^T x + b$, $\beta \in \mathbb{R}^d$, $b \in \mathbb{R}$
- **Norme :** $\|f\| = \|\beta\| = \sqrt{\beta^T \beta}$
- **Critère :** moindres carrés $(y - f(x))^2$
- **Problèmes :**

Problème de la régression

- Problème de prédiction d'une variable d'intérêt réelle à partir d'un vecteur $X \in \mathbb{R}^d$ de variables explicatives
- **Données :** $\{(x_i, y_i)\}_{i=1,\dots,n}$
- **Prédicteur :** $f(x) = \beta^T x + b$, $\beta \in \mathbb{R}^d$, $b \in \mathbb{R}$
- **Norme :** $\|f\| = \|\beta\| = \sqrt{\beta^T \beta}$
- **Critère :** moindres carrés $(y - f(x))^2$
- **Problèmes :**
  - overfitting quand d est grand,

Problème de la régression

- Problème de prédiction d'une variable d'intérêt réelle à partir d'un vecteur $X \in \mathbb{R}^d$ de variables explicatives
- **Données :** $\{(x_i, y_i)\}_{i=1,\dots,n}$
- **Prédicteur :** $f(x) = \beta^T x + b\ , \quad \beta \in \mathbb{R}^d,\ b \in \mathbb{R}$
- **Norme :** $\|f\| = \|\beta\| = \sqrt{\beta^T \beta}$
- **Critère :** moindres carrés $(y - f(x))^2$
- **Problèmes :**
  - overfitting quand d est grand,
  - sensibilité aux valeurs aberrantes.

Régression "ridge"

Idée : imposer une contrainte sur la norme L_2 des poids

Problème d'optimisation

$$\min_f \sum_{i=1}^{n} (y_i - f(x_i))^2$$

sous la contrainte:

$$\|f\|^2 \leq \lambda$$

Régression "ridge"

Formulation équivalente

$$\min_f \sum_{i=1}^{n} (y_i - f(x_i))^2 + C\|f\|^2$$

On note $X = (x_1, \ldots, x_n)^T$, $Y = (y_1, \ldots, y_n)^T$

Régression "ridge"

Formulation équivalente

$$\min_f \sum_{i=1}^{n} (y_i - f(x_i))^2 + C\|f\|^2$$

On note $X = (x_1, \ldots, x_n)^T$, $Y = (y_1, \ldots, y_n)^T$

Solution :

$$\hat{\theta} = (X^T X + C\ Id)^{-1} X^T Y$$

SVM pour la régression

On cherche le prédicteur sous la forme

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x)$$

On pose $K = (k(x_i, x_j))_{i,j}$ la matrice de Gram et on définit:

$$\|f\|_K^2 = \alpha^T K \alpha$$

SVM pour la régression

On cherche le prédicteur sous la forme

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x)$$

On pose $K = (k(x_i, x_j))_{i,j}$ la matrice de Gram et on définit:

$$\|f\|_K^2 = \alpha^T K \alpha$$

Problème d'optimisation

$$\min_{\alpha} \sum_{i=1}^{n} (y_i - K\alpha)^2 + C\alpha^T K \alpha$$

SVM pour la régression

On cherche le prédicteur sous la forme

$$f(x) = \sum_{i=1}^{n} \alpha_i k(x_i, x)$$

On pose $K = (k(x_i, x_j))_{i,j}$ la matrice de Gram et on définit:

$$\|f\|_K^2 = \alpha^T K \alpha$$

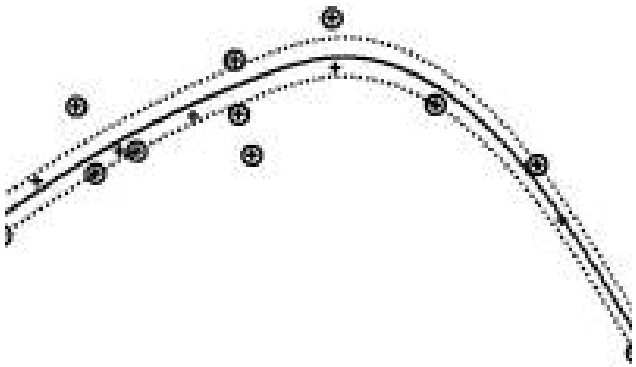
Problème d'optimisation

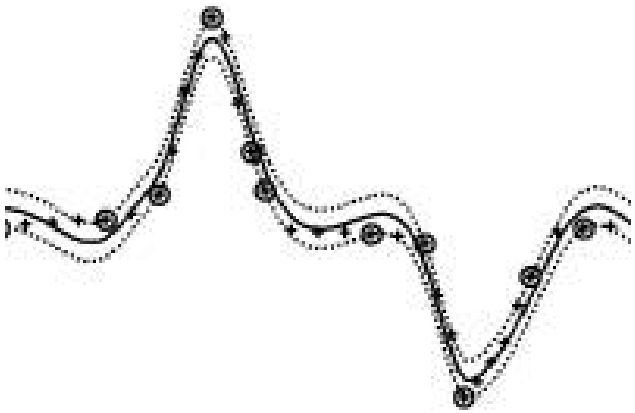
$$\min_{\alpha} \sum_{i=1}^{n} (y_i - K\alpha)^2 + C\alpha^T K \alpha$$

Solution :

$$\hat{\alpha} = (K + Cn\ Id)^{-1} Y$$

SVM pour la régression





$\Rightarrow$ concept analogue de vecteurs de support

Conclusion

On a vu ...

- Idées géométriques sous-jacentes à la conception des SVM

On a vu ...

- Idées géométriques sous-jacentes à la conception des SVM
- Aspects algorithmiques des SVM - optimisation

On a vu ...

- Idées géométriques sous-jacentes à la conception des SVM
- Aspects algorithmiques des SVM - optimisation
- Aspects statistiques - contrôle de la complexité

On a vu ...

- Idées géométriques sous-jacentes à la conception des SVM
- Aspects algorithmiques des SVM - optimisation
- Aspects statistiques - contrôle de la complexité
- Adaptation (sans douleur!) de méthodes statistiques classiques pour l'analyse de données

On a vu ...

- Idées géométriques sous-jacentes à la conception des SVM
- Aspects algorithmiques des SVM - optimisation
- Aspects statistiques - contrôle de la complexité
- Adaptation (sans douleur!) de méthodes statistiques classiques pour l'analyse de données
- SVM en action avec l'utilisation du logiciel SPIDER

On a tu ...

- Aspects fonctionnels dans l'étude des SVM (théorie des RKHS)

On a tu ...

- Aspects fonctionnels dans l'étude des SVM (théorie des RKHS)
- Technologie de construction de noyaux dédiés

On a tu ...

- Aspects fonctionnels dans l'étude des SVM (théorie des RKHS)
- Technologie de construction de noyaux dédiés
- Variantes des SVM

On a tu ...

- Aspects fonctionnels dans l'étude des SVM (théorie des RKHS)
- Technologie de construction de noyaux dédiés
- Variantes des SVM
- Autres applications des méthodes à noyaux : Kernel CCA, Kernel ICA, One-Class SVM, ... etc...

On a tu ...

- Aspects fonctionnels dans l'étude des SVM (théorie des RKHS)
- Technologie de construction de noyaux dédiés
- Variantes des SVM
- Autres applications des méthodes à noyaux : Kernel CCA, Kernel ICA, One-Class SVM, ... etc...
- Concurrents des SVM comme le boosting

Sources

- http://www.kernel-machines.org
- Vapnik (1998) - Statistical Learning Theory
- Hastie, Tibshirani, Friedman (2001) - The Elements of Statistical Learning
- Schölkopf, Tsuda, Vert (2004) - Kernel Methods in Computational Biology
- Autres ouvrages de référence:
  - Schölkopf, Burges, Smola (1998) - Advances in Kernel Methods: Support Vector Learning
  - Cristianini, Shawe-Taylor (2000) - An Introduction to Support Vector Machines and Other Kernel-based Learning Methods
  - Schölkopf, Smola (2002) - Learning with Kernels
  - Shawe-Taylor, Cristianini (2004) - Kernel Methods for Pattern Analysis