# Méthodes à noyaux

## *Cours Master 2004/05*

Jean-Philippe Vert

Jean-Philippe.Vert@mines.org

# Plan

- L'astuce noyau
- Le théorème du représentant
- Kernel PCA
- Régression par moindres carrés régularisés
- Support Vector Machines

# Introduction

La puissance des noyaux définis positifs sur un espace $\mathcal{X}$ provient de deux résultats ayant d'importantes implications pratiques:

- *l'astuce noyau* (*kernel trick*), basée sur la représentation d'un noyau d.p. comme un produit scalaire;
- le *théorème du représentant* (*representer theorem*) basé sur la fonctionnelle de régularisation définie par un noyau d.p.

Ces deux propriétés permettent le développement de nombreux algorithmes d'analyse: les *méthodes à noyau*.

# L'astuce noyau (*kernel trick*)

# Rappel

Nous avons démontré le théorème suivant

**Théorème 1** *Si $K$ est un n.d.p. sur un espace $\mathcal{X}$ quelconque, alors il existe un espace de Hilbert $\mathcal{H}$ muni du produit scalaire $\langle .,. \rangle_{\mathcal{H}}$ et une application*

$$\Phi : \mathcal{X} \mapsto \mathcal{H},$$

*tels que:*

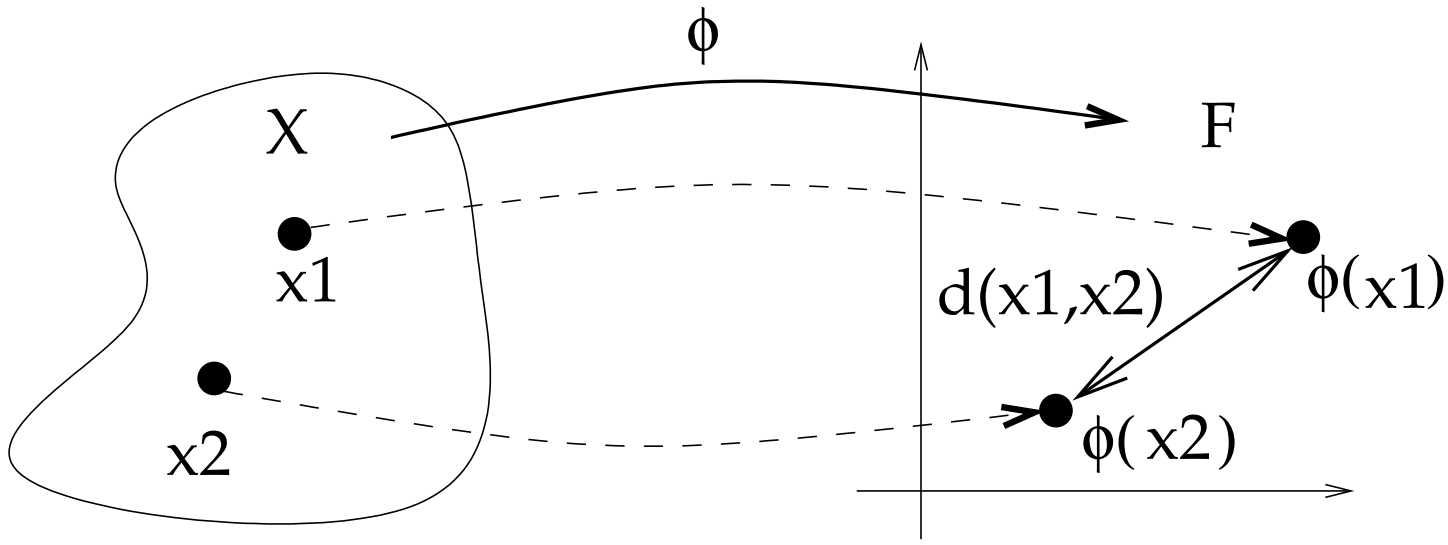
$$\forall \left(\mathbf{x}, \mathbf{x}'\right) \in \mathcal{X}^2, \quad K\left(\mathbf{x}, \mathbf{x}'\right) = \left\langle \Phi\left(\mathbf{x}\right), \Phi\left(\mathbf{x}'\right) \right\rangle_{\mathcal{H}}.$$

# L'astuce noyau

**Proposition 2** *Tout algorithme pour vecteurs qui puisse ne s'exprimer qu'en termes de produits scalaires entre vecteurs peut être effectué implicitement dans un espace de Hilbert en remplaçant chaque produit scalaire par l'évaluation d'un n.d.p. sur un espace quelconque.*

Ce théorème trivial a d'immenses implications pratiques.

# Exemple: calcul de distance




$$d\left(\mathbf{x}_1, \mathbf{x}_2\right)^2 = \| \Phi\left(\mathbf{x}_1\right) - \Phi\left(\mathbf{x}_2\right) \|^2$$
$$= \left(\Phi\left(\mathbf{x}_1\right) - \Phi\left(\mathbf{x}_2\right)\right).\left(\Phi\left(\mathbf{x}_1\right) - \Phi\left(\mathbf{x}_2\right)\right)$$
$$= \Phi\left(\mathbf{x}_1\right).\Phi\left(\mathbf{x}_1\right) + \Phi\left(\mathbf{x}_2\right).\Phi\left(\mathbf{x}_2\right) - 2\Phi\left(\mathbf{x}_1\right).\Phi\left(\mathbf{x}_2\right)$$

$$d(\mathbf{x}_1, \mathbf{x}_2)^2 = K(\mathbf{x}_1, \mathbf{x}_1) + K(\mathbf{x}_2, \mathbf{x}_2) - 2K(\mathbf{x}_1, \mathbf{x}_2)$$

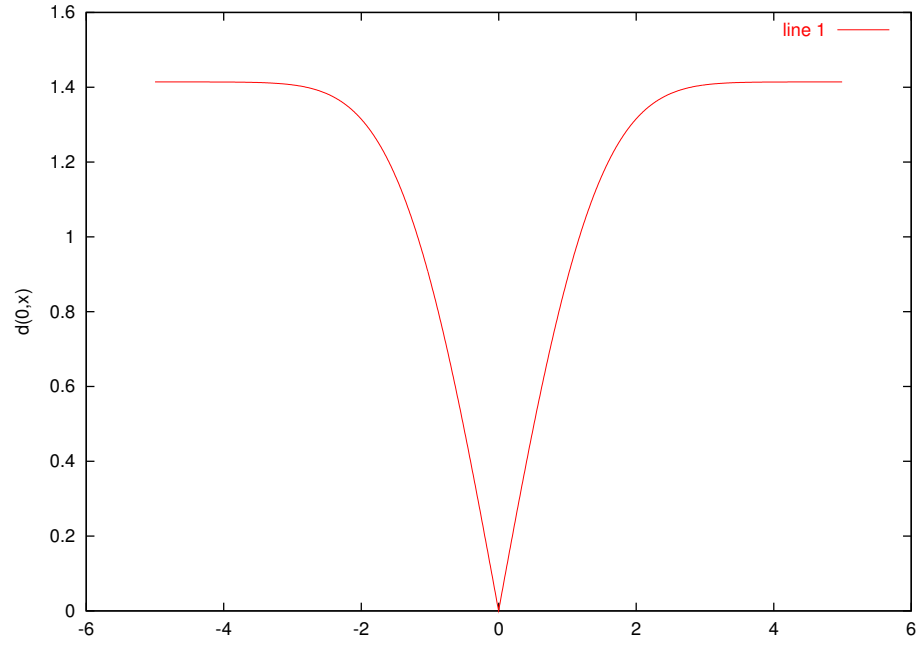
# Calcul de distance

Example:

$$K(\mathbf{x},\mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}},$$

alors

$$d(\mathbf{x},\mathbf{y}) = \sqrt{2\left[1 - e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}\right]}$$



# Exemple: distance point - ensemble

- Soit $\mathcal{S} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$ un ensemble fini d'objets
- Soit $\mathbf{x}$ un point quelconque
- Comment mesurer la *distance entre* $\mathbf{x}$ *et* $\mathcal{S}$?

# Données vectorielles

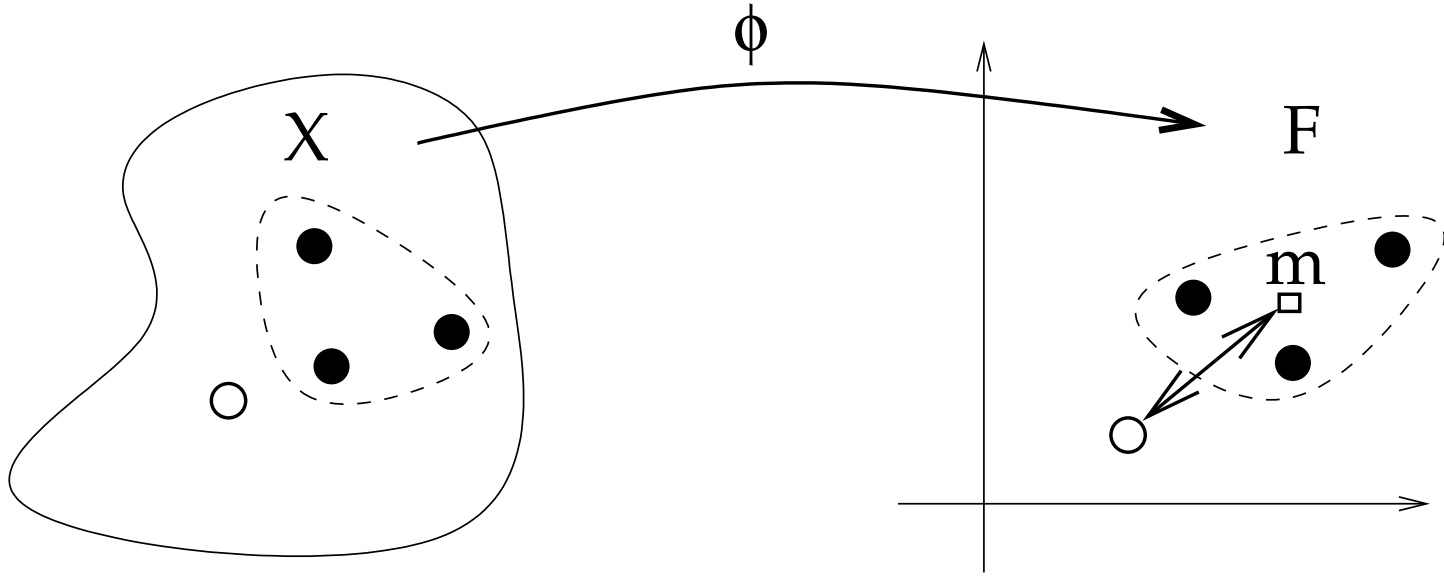
Si $\mathbf{x}_i \in \mathbb{R}^d$, alors on peut définir le barycentre de $\mathcal{S}$:

$$m := \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i,$$

et définir la distance de $\mathbf{x}$ à $\mathcal{S}$ par:

$$d(\mathbf{x}, \mathcal{S}) := \| \mathbf{x} - m \|$$

# Astuce noyau




$$d\left(\mathbf{x},\mathcal{S}\right) = \| \Phi\left(\mathbf{x}\right) - \frac{1}{n}\sum_{i=1}^{n} \Phi(\mathbf{x}_i) \|$$

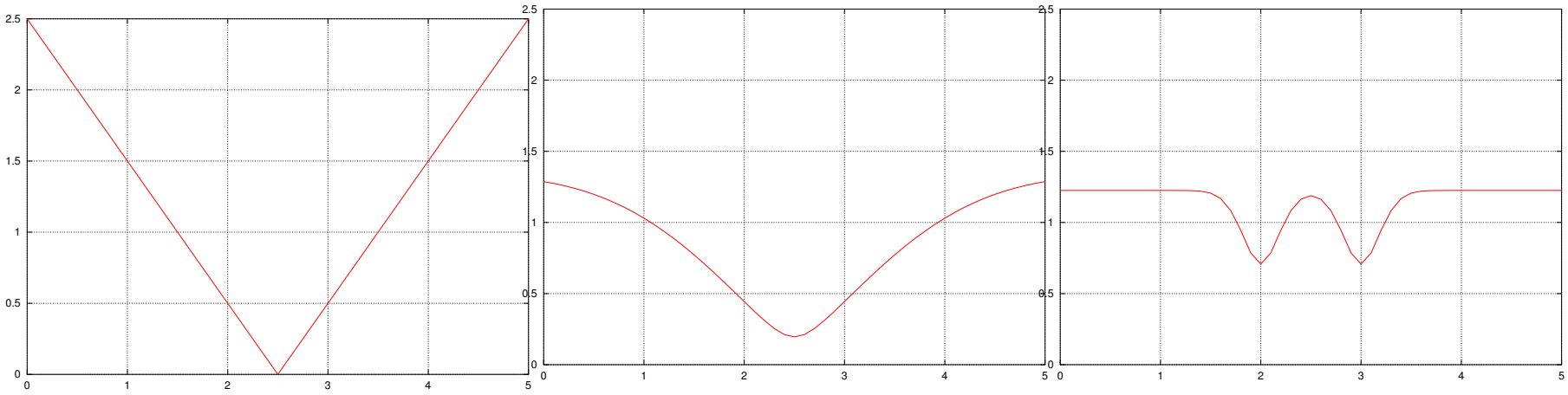
$$= \sqrt{K(\mathbf{x},\mathbf{x}) - \frac{2}{n}\sum_{i=1}^{n} K(\mathbf{x},\mathbf{x}_i) + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} K(\mathbf{x}_i,\mathbf{x}_j)}.$$

# Remarque

- Le point $m$ n'a *pas nécessairement de pré-image* dans $\mathcal{X}$
- La distance obtenue est une distance Euclidienne

# Exemples 1D

Distances avec l'ensemble de deux points $\mathcal{S} = \{2, 3\}$ pour différents noyaux:



$K(\mathbf{x}, \mathbf{y}) = \mathbf{xy}.$

(linéaire)

$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}.$

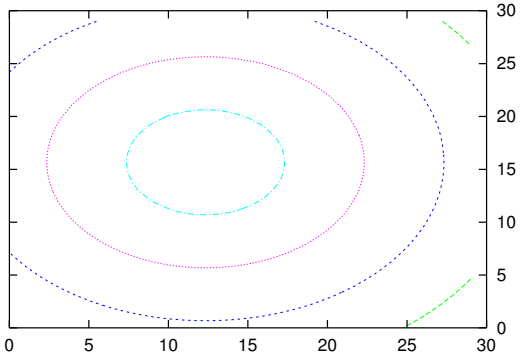
avec $\sigma = 1.$

$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}.$

avec $\sigma = 0.2.$

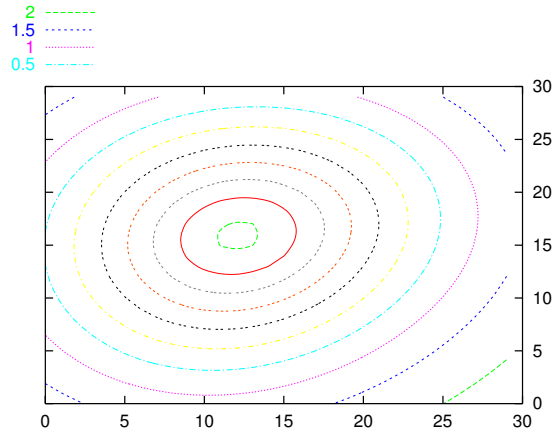
# Exemples 2D

Distances avec l'ensemble de $3$ points
$\mathcal{S} = \{(1,1)', (1,2)', (2,2)'\}$ pour différents noyaux:



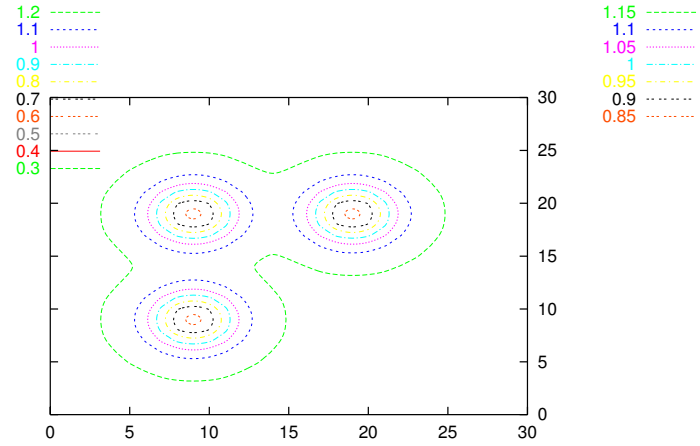

$K(\mathbf{x}, \mathbf{y}) = \mathbf{x}\mathbf{y}.$

(linéaire)




$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}.$

avec $\sigma = 1.$




$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}.$

avec $\sigma = 0.2.$

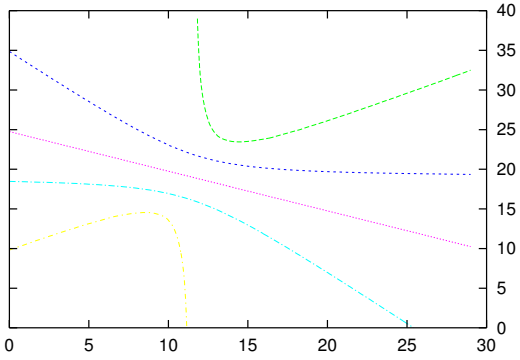
# Application en discrimination

- Soient $\mathcal{S}_1$ et $\mathcal{S}_2$ deux ensembles d'objets appartenant à 2 classes différentes. Soit $\mathbf{x}$ un nouvel objet. Quel est sa classe?
- On peut choisir la classe $\mathcal{S}_i$ qui minimise $d(\mathbf{x}, \mathcal{S}_i)$.
- Grâce aux formules précédentes, on obtient:

$$d(\mathbf{x}, \mathcal{S}_1)^2 - d(\mathbf{x}, \mathcal{S}_2)^2 = \frac{1}{|\mathcal{S}_1|} \sum_{\mathbf{x}_i \in \mathcal{S}_1} K(\mathbf{x}, \mathbf{x}_i) - \frac{1}{|\mathcal{S}_2|} \sum_{\mathbf{x}_j \in \mathcal{S}_2} K(\mathbf{x}, \mathbf{x}_j) + b.$$

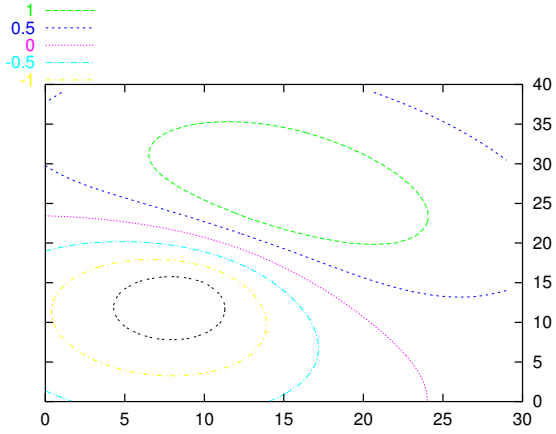
# Exemple de discrimination

Courbes de niveaux de $d(\mathbf{x}, \mathcal{S}_1) - d(\mathbf{x}, \mathcal{S}_2)$ pour les ensembles $\mathcal{S}_1 = \{(1,1)', (1,2)'\}$ et $\mathcal{S}_1 = \{(1,3)', (2,2)'\}$ pour différents noyaux:



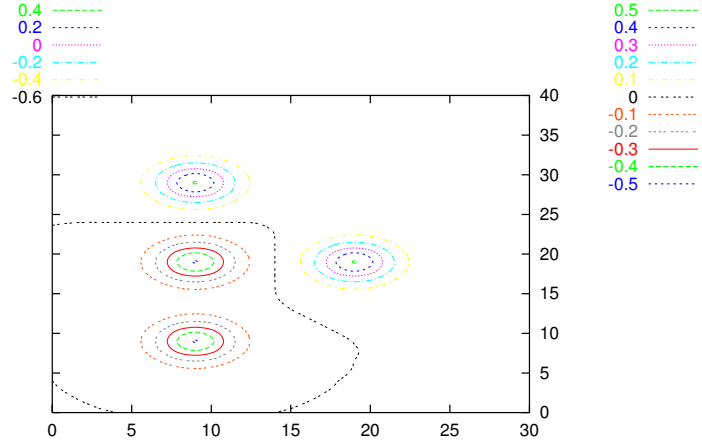

$K(\mathbf{x}, \mathbf{y}) = \mathbf{x}\mathbf{y}$.

(linéaire)




$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}$.

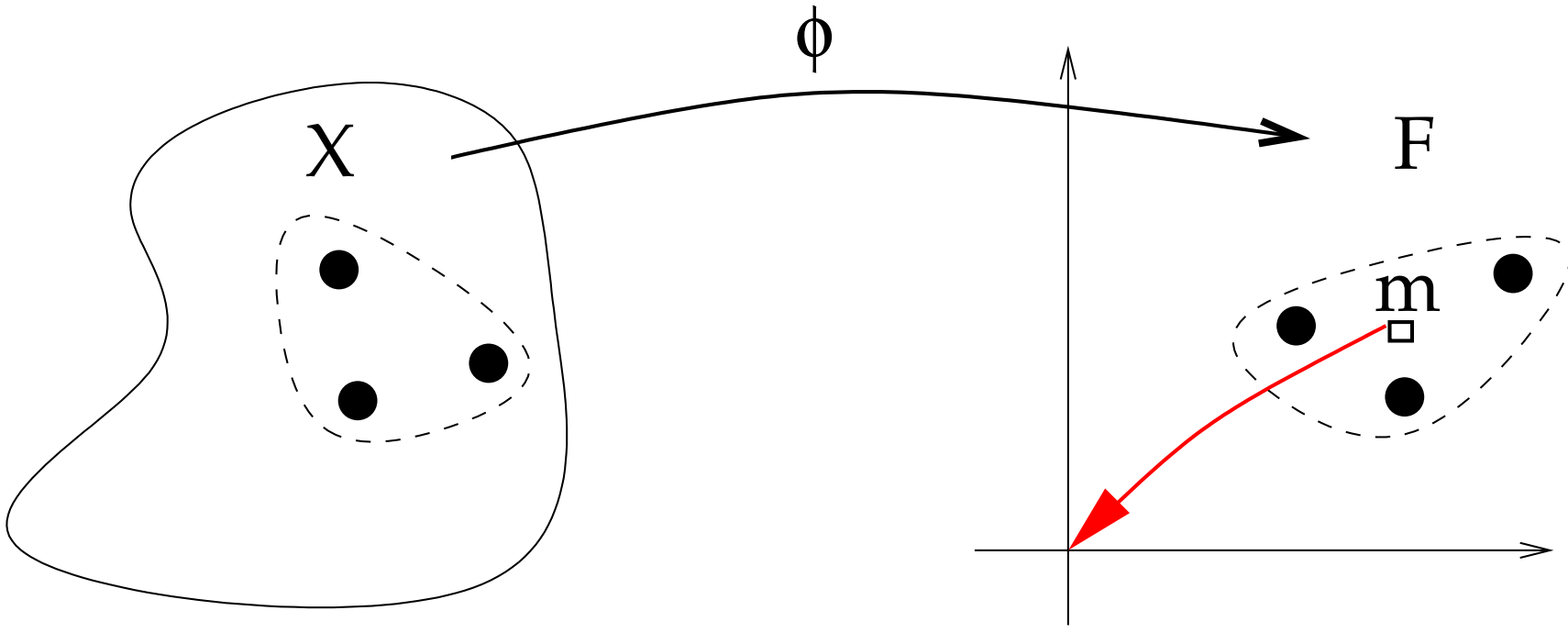
avec $\sigma = 1$.




$K(\mathbf{x}, \mathbf{y}) = e^{-\frac{(\mathbf{x}-\mathbf{y})^2}{2\sigma^2}}$.

avec $\sigma = 0.2$.

# Exemple: centrer des données




Soit $K$ une matrice de Gram $n \times n$. Comment calculer la matrice de Gram $n \times n$ pour les données centrées par translation?

# Exemple: centrer des données (cont)

On calcule, pour $0 \leq i, j \leq n$:

$$\begin{aligned} K'_{i,j} &= \left(\Phi\left(\mathbf{x}_i\right) - m\right).\left(\Phi\left(\mathbf{x}_j\right) - m\right) \\ &= \Phi\left(\mathbf{x}_i\right).\Phi\left(\mathbf{x}_j\right) - m.\left(\Phi\left(\mathbf{x}_i\right) + \Phi\left(\mathbf{x}_j\right)\right) + m.m \\ &= K_{i,j} - \frac{1}{n}\sum_{k=1}^{n}\left(K_{i,k} + K_{j,k}\right) + \frac{1}{n^2}\sum_{k,l=1}^{n} K_{k,l} \end{aligned}$$

donc:

$$K' = K - UK - KU + UKU = \left(I - U\right)K\left(I - U\right),$$

avec $U_{i,j} = 1/n$ pour $1 \leq i, j \leq n$.

# Bilan

Grâce à l'astuce noyau, il est possible de:

- Rendre *non-linéaire* des méthodes linéaires (avec un noyau RBF Gaussien par exemple)
- Plonger l'espace initial dans un *espace plus grand* et y travailler implicitement (par exemple, trouver un barycentre n'ayant pas de pré-image)

# Le théorème du représentant

# Le théorème (Kimeldorf et Wahba, 1970)

**Théorème 3** *Soit $\mathcal{X}$ un ensemble muni d'un noyau d.p. $K$, $\mathcal{H}_K$ le rkhs associé, et $\mathcal{S} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\} \subset \mathcal{X}$ un ensemble fini d'objets.*
*Soit $\Psi : \mathbb{R}^{n+1} \to \mathbb{R}$ une fonction de $n+1$ arguments, strictement croissante par rapport au dernier argument. Alors, toute solution au problème:*

$$\min_{f \in \mathcal{H}_K} \Psi\left(f\left(\mathbf{x}_1\right), \cdots, f\left(\mathbf{x}_n\right), \| f \|_{\mathcal{H}_K}\right), \tag{1}$$

*admet une représentation de la forme:*

$$\forall \mathbf{x} \in \mathcal{X}, \quad f\left(\mathbf{x}\right) = \sum_{i=1}^{n} \alpha_i K\left(\mathbf{x}_i, \mathbf{x}\right). \tag{2}$$

# Remarques

Souvent, la fonction $\Psi$ a la forme:

$$\Psi\left(f\left(\mathbf{x}_1\right),\cdots,f\left(\mathbf{x}_n\right),\parallel f\parallel_{\mathcal{H}_K}\right)=c\left(f\left(\mathbf{x}_1\right),\cdots,f\left(\mathbf{x}_n\right)\right)+\lambda\Omega\left(\parallel f\parallel_{\mathcal{H}}\right.$$

où $c(.)$ mesure le 'fit' de $f$ à un problème. Cette formulation a deux effets:

- La solution aura une *norme* $\parallel f\parallel_{\mathcal{H}_K}$ *aussi petite que possible*, ce qui peut etre bénéfique en soit (par exemple, si on recherche une fonction régulière)
- En plus, par le théorème précédent, on sait d'avance que la solution sera dans un *sous-espace de dimension* $n$ *connue* (bien que $\mathcal{H}_K$ puisse être de dimension infinie), ce qui permet de développer des *algorithmes efficaces*.

# Remarques (cont)

La plupart des méthodes à noyaux ont deux interprétations complémentaires:

- une interprétation *géométrique* dans le feature space, grâce à l'astuce noyau. Meme si le feature space est grand, nous travaillerons en général dans le sous-espace de dimension au plus $s$ engendré par les $n$ points donnés.
- une interprétation *fonctionnelle*, grâce au théorème du représentant. C'est lui qui assure que nous travaillons en dimension finie.

Bien sûr ces 2 interprétations se superposent quand on voit le rkhs comme un feature space fonctionnel.

# Preuve

- Soit $\xi(f, \mathcal{S})$ la fonction à minimiser dans le théorème, et

$$\mathcal{H}_K^{\mathcal{S}} = \left\{ f \in \mathcal{H}_K : f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}), (\alpha_1, \cdots, \alpha_n) \in \mathbb{R}^n \right\}$$

- $\mathcal{H}_K^{\mathcal{S}}$ est un sous-espace de dimension finie, donc toute fonction $f \in \mathcal{H}_K$ est décomposable de manière unique en:

$$f = f_{\mathcal{S}} + f_{\perp},$$

avec $f_{\mathcal{S}} \in \mathcal{H}_K^{\mathcal{S}}$ et $f_{\perp} \perp \mathcal{H}_K^{\mathcal{S}}$ (projection orthogonale).

# Preuve (cont.)

- $\mathcal{H}_K$ étant un rkhs, on a donc:

$$\forall i = 1, \cdots, n, \quad f_\perp(\mathbf{x}_i) = \langle f_\perp, K(\mathbf{x}_i, .)\rangle_{\mathcal{H}_K} = 0$$

car $K(\mathbf{x}_i, .) \in \mathcal{H}_K$, et donc:

$$\forall i = 1, \cdots, n, \quad f(\mathbf{x}_i) = f_{\mathcal{S}}(\mathbf{x}_i).$$

- Le théorème de Pythagore dans $\mathcal{H}_K$ montre que:

$$\| f \|^2_{\mathcal{H}_K} = \| f_{\mathcal{S}} \|^2_{\mathcal{H}_K} + \| f_\perp \|^2_{\mathcal{H}_K}.$$

# Preuve (cont.)

- On en déduit que:

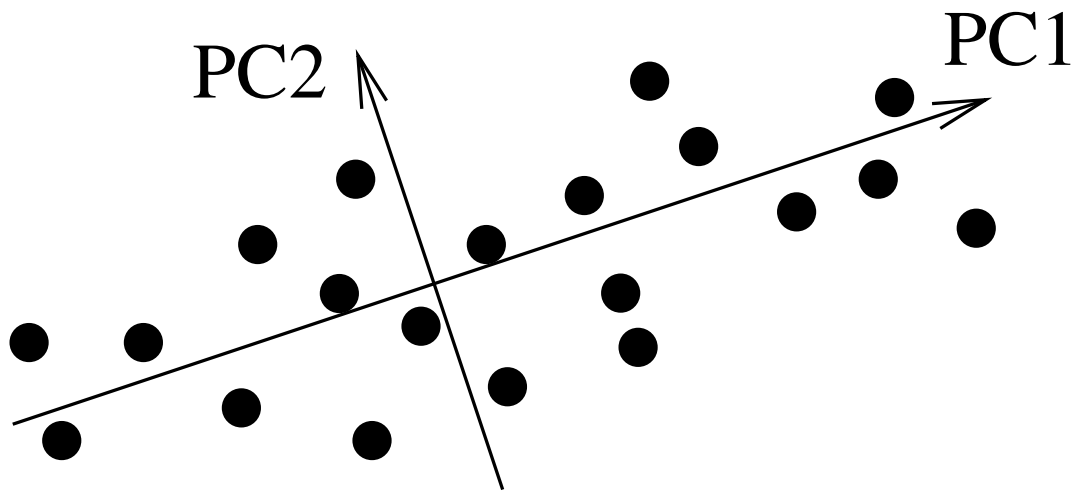
$$\xi\left(f, \mathcal{S}\right) \geq \xi\left(f_{\mathcal{S}}, \mathcal{S}\right),$$

avec égalité si et seulement si $\| f_{\perp} \|_{\mathcal{H}_K} = 0$. Le minimum de $\Psi$ est donc nécessairement dans $\mathcal{H}_K^{\mathcal{S}}$. $\square$

# ACP à noyau

# ACP: rappels

- Soit $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ un ensemble de vecteurs ($\mathbf{x}_i \in \mathbb{R}^d$)
- L'*analyse en composantes principales* (ACP) permet de détecter les directions de forte variation



# ACP classique

- Supposons que les données sont *centrées*:

$$\sum_{i=1}^{n} \mathbf{x}_i = 0.$$

- La *projection orthogonale* sur une direction $\mathbf{w} \in \mathbb{R}^d$ est la fonction $h_{\mathbf{w}} : \mathcal{X} \to \mathbb{R}$ définie par

$$h_{\mathbf{w}}(\mathbf{x}) = \mathbf{x}^\top \frac{\mathbf{w}}{\| \mathbf{w} \|}.$$

# ACP: rappels (cont.)

- La *variance empirique* de $h_{\mathbf{w}}$ est:

$$\hat{var}\left(h_{\mathbf{w}}\right) := \frac{1}{n}\sum_{i=1}^{n} h_{\mathbf{w}}\left(\mathbf{x}_i\right)^2 = \frac{1}{n}\sum_{i=1}^{n} \frac{\left(\mathbf{x}_i^\top \mathbf{w}\right)^2}{\| \mathbf{w} \|^2}.$$

- La $i$-eme direction principale $w_i$ ($i = 1, \ldots, d$) est définie par:

$$\mathbf{w}_i = \underset{\mathbf{w} \perp \{\mathbf{w}_1, \ldots, \mathbf{w}_{i-1}\}}{\arg\max} \hat{var}\left(h_{\mathbf{w}}\right).$$

# ACP: rappels (cont.)

- Soit $X$ la matrice $n \times d$ dont les lignes sont les données $\mathbf{x}_1, \ldots, \mathbf{x}_n$. On a alors:

$$\hat{var}\left(h_{\mathbf{w}}\right) = \frac{1}{n} \sum_{i=1}^{n} \frac{\left(\mathbf{x}_i^\top \mathbf{w}\right)^2}{\| \mathbf{w} \|^2} = \frac{1}{n} \frac{\mathbf{w}^\top X^\top X \mathbf{w}}{\mathbf{w}^\top \mathbf{w}}.$$

- Les solutions de

$$\mathbf{w}_i = \underset{\mathbf{w} \perp \{\mathbf{w}_1, \ldots, \mathbf{w}_{i-1}\}}{\arg\max} \frac{1}{n} \frac{\mathbf{w}^\top X^\top X \mathbf{w}}{\mathbf{w}^\top \mathbf{w}}$$

sont les *vecteurs propres de* $C = X^\top X$, rangés par valeurs propres décroissantes

# Vision fonctionnelle

- Soit $K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{y}$ le noyau linéaire
- Le rkhs $\mathcal{H}$ associé est l'ensemble des fonctions:

$$f_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^\top \mathbf{x},$$

  muni de la norme $\| f_{\mathbf{w}} \|_{\mathcal{H}} = \| \mathbf{w} \|_{\mathbb{R}^d}$.
- On a donc:

$$\hat{var}(h_{\mathbf{w}}) = \frac{1}{n} \sum_{i=1}^{n} \frac{\left(\mathbf{x}_i^\top \mathbf{w}\right)^2}{\| \mathbf{w} \|^2} = \frac{1}{n \| f_{\mathbf{w}} \|^2} \sum_{i=1}^{n} f_{\mathbf{w}}(\mathbf{x}_i)^2.$$

- De plus, $\mathbf{w} \perp \mathbf{w}' \Leftrightarrow f_{\mathbf{w}} \perp f_{\mathbf{w}'}$

# Vision fonctionnelle (cont.)

- L'ACP consiste donc à résoudre, pour $i = 1, \ldots, d$:

$$f_i = \underset{f \perp \{f_1, \ldots, f_{i-1}\}}{\arg\max} \frac{1}{n \| f \|^2} \sum_{i=1}^{n} f(\mathbf{x}_i)^2.$$

- On peut appliquer le théorème du représentant (aussi valable dans un sous-espace): pour $i = 1, \ldots, d$, on a :

$$\forall \mathbf{x} \in \mathcal{X}, \quad f_i(\mathbf{x}) = \sum_{j=1}^{n} \alpha_{i,j} K(\mathbf{x}_j, \mathbf{x}),$$

avec $\boldsymbol{\alpha}_i = (\alpha_{i,1}, \ldots, \alpha_{i,n})^\top \in \mathbb{R}^n$.

# Vision fonctionnelle (cont)

On a donc:

$$\| f_i \|_{\mathcal{H}}^2 = \sum_{k,l=1}^{d} \alpha_{i,k} \alpha_{i,l} K(\mathbf{x}_k, \mathbf{x}_l) = \boldsymbol{\alpha}_i^\top K \boldsymbol{\alpha}_i,$$

et de même:

$$\sum_{k=1}^{n} f_i(\mathbf{x}_k)^2 = \boldsymbol{\alpha}_i^\top K^2 \boldsymbol{\alpha}_i.$$

# Vision fonctionnelle (cont)

Le problème devient donc de maximiser en $\boldsymbol{\alpha}$ la fonction:

$$\boldsymbol{\alpha}_i = \arg\max_{\boldsymbol{\alpha}} \frac{\boldsymbol{\alpha}^\top K^2 \boldsymbol{\alpha}}{n \boldsymbol{\alpha}^\top K \boldsymbol{\alpha}},$$

sous les contraintes $\boldsymbol{\alpha}_i^\top K \boldsymbol{\alpha}_j = 0$ pour $j = 1, \ldots, i-1$.

# Vision fonctionnelle (cont)

- Soient $(e_1, \ldots, e_n)$ une base orthonormale de vecteurs propres de $K$ associés aux valeurs propres $\lambda_1 \geq \ldots \geq \lambda_n \geq 0$.
- Soit $\boldsymbol{\alpha}_i = \sum_{j=1}^n \beta_{ij} e_j$, alors

$$\frac{\boldsymbol{\alpha}^\top K^2 \boldsymbol{\alpha}}{n \boldsymbol{\alpha}^\top K \boldsymbol{\alpha}} = \frac{\sum_{i=1}^n \beta_{ii}^2 \lambda_i^2}{n \sum_{i=1}^n \beta_{ii}^2 \lambda_i},$$

qui est maximum pour $\boldsymbol{\alpha}_1 = \beta_{11} e_1$, $\boldsymbol{\alpha}_2 = \beta_{22} e_2$, *etc...*

# Normalisation

- Pour $\boldsymbol{\alpha}_i = \beta_{ii} e_i$, on veut:

$$1 = \| f_i \|^2_{\mathcal{H}} = \boldsymbol{\alpha}_i^\top K \boldsymbol{\alpha}_i = \beta_{ii}^2 \lambda_i,$$

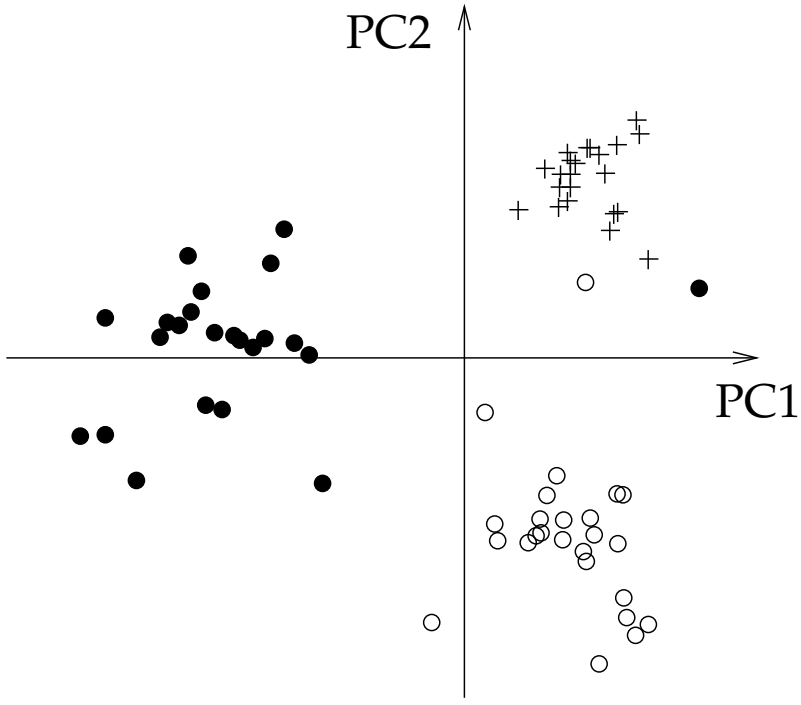
donc:

$$\boldsymbol{\alpha}_i = \frac{1}{\sqrt{\lambda_i}} e_i.$$

# Remarques

- Il faut diagonaliser la matrice de Gram (centrée) au lieu de la matrice de covariance.
- L'analyse est identique en remplacant le noyau linéaire par n'importe quel noyau d.p.

# Exemple

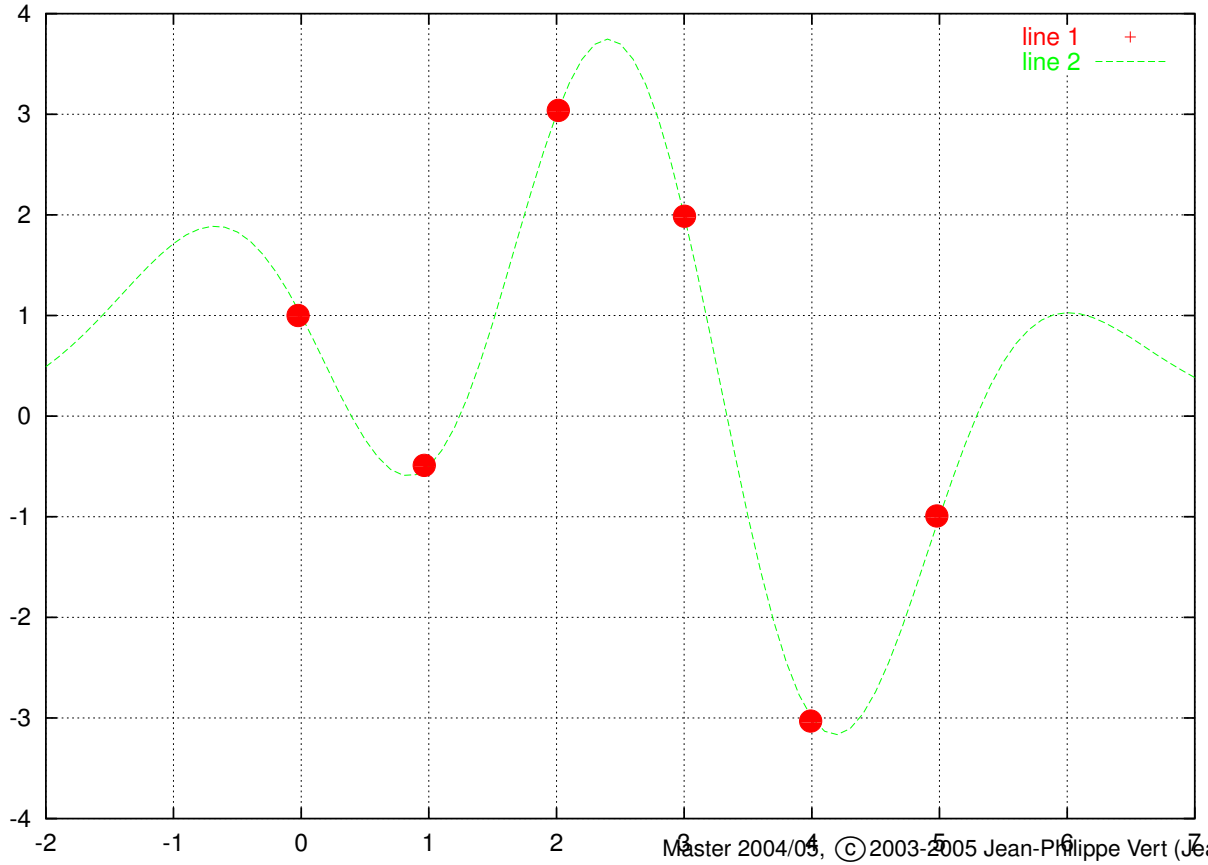



A set of 74 human tRNA sequences is analyzed using a kernel for sequences (the second-order marginalized kernel based on SCFG). This set of tRNAs contains three classes, called Ala-AGC (*white circles*), Asn-GTT (*black circles*) and Cys-GCA (*plus symbols*) (from Tsuda et al., 2003).

# Régression par moindres carrés régularisés

# Régression

- Soit $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{X}^n$ un ensemble de points
- Soit $\{y_1, \ldots, y_n\} \in \mathbb{R}^n$ des nombres associés aux points
- Régression = trouver $f : \mathcal{X} \to \mathbb{R}$ pour *prédire* $y$ *par* $f(\mathbf{x})$



# Moindres carrés

- On mesure l'erreur si $f$ prédit $f(\mathbf{x})$ au lieu de $y$ par:

$$V(f(\mathbf{x}), y) = (y - f(\mathbf{x}))^2 .$$

- On se fixe un ensemble de fonctions $\mathcal{H}$
- La *régression par moindres carrés* consiste à minimiser:

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i))^2$$

- Problèmes: instabilité, risque d'overfitting si $\mathcal{H}$ est grand

# Moindres carrés régularisés

- On choisit $\mathcal{H} = \mathcal{H}_K$, le rkhs associé à un noyau $K$ sur $\mathcal{X}$
- On *régularise* la fonctionnelle à minimiser par:

$$\hat{f} = \arg\min_{f \in \mathcal{H}_K} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - f\left(\mathbf{x}_i\right)\right)^2 + \lambda \| f \|_{\mathcal{H}_K}^2 .$$

- *1er effet =* prévenir l'overfitting *en pénalisant les fonctions trop 'irrégulières' (au sens du rkhs)*

# Représentation des solutions

Par le *théorème du représentant*, toute solution de

$$\hat{f} = \arg\min_{f \in \mathcal{H}_K} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - f\left(\mathbf{x}_i\right)\right)^2 + \lambda \| f \|_{\mathcal{H}_K}^2 .$$

peut s'écrire sous la forme:

$$\hat{f} = \sum_{i=1}^{n} \alpha_i K\left(\mathbf{x}_i, \mathbf{x}\right) .$$

*2e effet = simplifie énormément la solution*

# Ecriture duale

- Soit $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top \in \mathbb{R}^n$,
- $K$ la matrice de Gram $n \times n$: $K_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$ .
- On a alors:

$$\left(\hat{f}(\mathbf{x}_1), \ldots, \hat{f}(\mathbf{x}_n)\right)^\top = K\boldsymbol{\alpha},$$

- et toujours

$$\| \hat{f} \|^2_{\mathcal{H}_K} = \boldsymbol{\alpha}^\top K \boldsymbol{\alpha}.$$

# Ecriture duale (cont.)

Le problème est donc équivalent à:

$$\arg\min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \frac{1}{n} \left(K\boldsymbol{\alpha} - y\right)^\top \left(K\boldsymbol{\alpha} - y\right) + \lambda \boldsymbol{\alpha}^\top K \boldsymbol{\alpha}.$$

C'est une fonction convexe et différentiable en $\boldsymbol{\alpha}$: il est équivalent d'annuler sa dérivée:

$$\begin{aligned} 0 &= \frac{2}{n} K \left(K\boldsymbol{\alpha} - y\right) + 2\lambda K \boldsymbol{\alpha} \\ &= K \left[\left(K + \lambda n I\right) \boldsymbol{\alpha} - y\right] \end{aligned}$$

# Ecriture duale (cont.)

- $K$ étant une matrice symétrique elle est diagonalisable en base orthonormée, avec $Ker(K) \perp Im(K)$
- Dans cette base on voit que $(K + \lambda n I)^{-1}$ laisse $Im(K)$ et $Ker(K)$ invariants
- Donc le problème est équivalent à:

$$(K + \lambda n I)\,\boldsymbol{\alpha} - y \in Ker(K)$$

$$\Leftrightarrow \boldsymbol{\alpha} - (K + \lambda n I)^{-1}\, y \in Ker(K)$$

$$\Leftrightarrow \boldsymbol{\alpha} = (K + \lambda n I)^{-1}\, y + \epsilon, \text{ avec } K\epsilon = 0.$$

# Ecriture duale (cont.)

- Mais si $\boldsymbol{\alpha}' = \boldsymbol{\alpha} + \epsilon$ avec $K\epsilon = 0$, alors:

$$\| f - f' \|^2_{\mathcal{H}_K} = \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}'\right)^\top K \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}'\right) = 0,$$

  donc $f = f'$.

- La solution au problème initial est donc par exemple:

$$\hat{f} = \sum_{i=1}^{n} \alpha_i K\left(\mathbf{x}_i, \mathbf{x}\right),$$

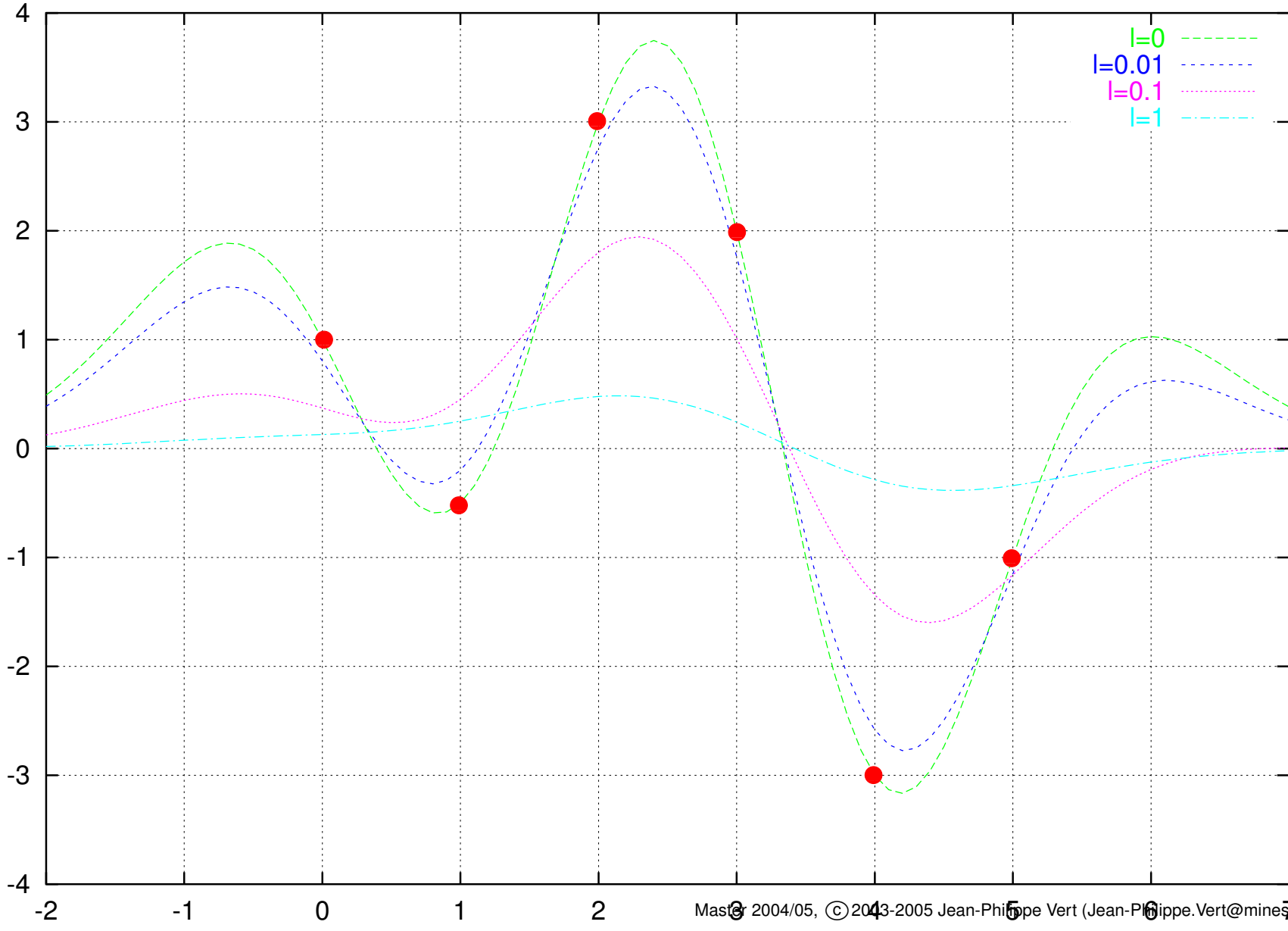
  avec

$$\boldsymbol{\alpha} = \left(K + \lambda n I\right)^{-1} y.$$

# Remarques

- La matrice $(K + n\lambda I)^{-1}$ est *inversible quand* $\lambda > 0$.
- Quand $\lambda \to 0$, la méthode converge vers les moindres carrés ordinaires (minimisation de la perte empirique). Quand $\lambda \to \infty$, la solution tend vers $f = 0$.
- En pratique, on n'inverse pas la matrice $K + n\lambda I$, on utilise des *algorithmes de résolution de systèmes linéaire*.
- Cette méthode pose des problèmes pratiques quand *le nombre de points augmente*.

# Exemple



# Support Vector Machines (SVM) pour la classification binaire

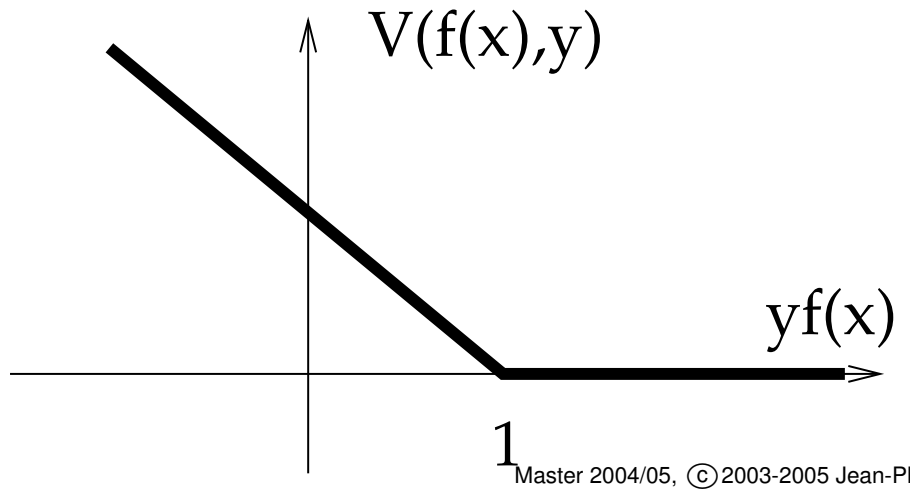
# Classification binaire

- Soit $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathcal{X}^n$ un ensemble de points
- Soit $\{y_1, \ldots, y_n\} \in \{-1, +1\}$ des labels binaires associés aux points
- Classification = trouver $f : \mathcal{X} \to \{-1, +1\}$ pour *prédire le label $y$ par $f(\mathbf{x})$*
- Ce problème est souvent appelé *reconnaissance de forme* (*pattern recognition*).

# La *hinge loss*

- On va chercher une fonction $f : \mathcal{X} \to \mathbb{R}$, et la prédiction sera le signe de $f(\mathbf{x})$.
- Si on prédit $f(\mathbf{x}) \in \mathbb{R}$ au lieu de $y \in \{-1, +1\}$, on va utiliser la fonction de perte suivante (*hinge loss*):

$$V_{hinge}(f(\mathbf{x}), y) = \begin{cases} 0 & \text{si } yf(\mathbf{x}) \geq 1, \\ 1 - yf(\mathbf{x}) & \text{sinon.} \end{cases}$$



# Pourquoi la hinge loss?

- Pourquoi pas $1$ si $yf(\mathbf{x}) < 0$, $0$ sinon (indicatrice d'erreur)?
- La hinge loss est une fonction convexe de $f$, ce qui va permettre d'aboutir à un *algorithme efficace* (minimiser une fonction convexe)
- Elle est minorée par la fonction erreur
- Elle force les points à s'éloigner de la frontière de discrimination (important pour la généralisation)

# Définition des SVM

- La SVM est l'algorithme qui consiste à minimiser

$$\hat{f} = \arg\min_{f \in \mathcal{H}_K} \frac{1}{n} \sum_{i=1}^{n} V_{hinge}\left(f\left(\mathbf{x}_i\right), y_i\right) + \lambda \| f \|_{\mathcal{H}_K}^2 .$$

ou $\mathcal{H}_K$ est le rkhs associé à un noyau $K$ sur $\mathcal{X}$.

- Par le théorème du représentant, on sait que $\hat{f}$ va se décomposer en

$$\hat{f}\left(\mathbf{x}\right) = \sum_{i=1}^{n} \alpha_i K\left(\mathbf{x}_i, \mathbf{x}\right).$$

# Variables "ressorts" (*slack variables*)

- La hinge loss est convexe en $f$ mais *non différentiable*. On ne peut donc pas résoudre directement le problème par dérivation
- On simplifie le problème en introduisant des variables ressort $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_n) \in \mathbb{R}^n$ et considérant le problème équivalent:

$$\min_{f \in \mathcal{H}_K, \boldsymbol{\xi} \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} \xi_i + \lambda \| f \|_{\mathcal{H}_K}^2,$$

sous les contraintes:

$$\xi_i \geq V\left(f\left(\mathbf{x}_i\right), y_i\right), \text{ pour } i = 1, \ldots, n.$$

# Variables ”ressorts” (cont.)

Ce problème est lui-meme équivalent à:

$$\min_{f \in \mathcal{H}_K, \boldsymbol{\xi} \in \mathbb{R}^n} \frac{1}{n} \sum_{i=1}^{n} \xi_i + \lambda \| f \|^2_{\mathcal{H}_K},$$

sous les contraintes:

$$\begin{cases} \xi_i \geq 1 - y_i f(\mathbf{x}_i), & \textit{pour } i = 1, \ldots, n, \\ \xi_i \geq 0, & \textit{pour } i = 1, \ldots, n, \end{cases}$$

# Problème primal

En remplacant $\hat{f}$ par

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}),$$

on obtient un problème en $\boldsymbol{\alpha}$:

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^n, \boldsymbol{\xi}\in\mathbb{R}^n} \frac{1}{n}\sum_{i=1}^{n} \xi_i + \lambda \boldsymbol{\alpha}^\top K \boldsymbol{\alpha},$$

sous les contraintes:

$$\begin{cases} y_i \sum_{j=1}^{n} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + \xi_i - 1 \geq 0, & \text{pour } i = 1, \ldots, n, \\ \xi_i \geq 0, & \text{pour } i = 1, \ldots, n. \end{cases}$$

# Résolution

- C'est un problème classique de minimiser une fonctionnelle quadratique sous des contraintes linéaires (*programme quadratique*).
- On pourrait utiliser directement des logiciels d'optimisation généraux
- Nous allons cependant le ré-écrire sous une forme duale pour mieux comprendre et accélérer l'algorithme

# Lagrangien

Introduisons le *Lagrangien* du problème, en utilisant les multiplicateurs de Lagrange $\boldsymbol{\mu} \geq 0$ et $\boldsymbol{\nu} \geq 0$:

$$L\left(\boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\nu}\right) = \frac{1}{n} \sum_{i=1}^{n} \xi_i + \lambda \boldsymbol{\alpha}^\top K \boldsymbol{\alpha}$$

$$- \sum_{i=1}^{n} \mu_i \left[ y_i \sum_{j=1}^{n} \alpha_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) + \xi_i - 1 \right] - \sum_{i=1}^{n} \nu_i \xi_i .$$

# Rappel (Lagrangien)

- Pour chaque $\mu \geq 0$ et $\nu \geq 0$, il faut *minimiser* en $\alpha$ et $\xi$:

$$\left(\alpha^*\left(\mu,\nu\right),\xi^*\left(\mu,\nu\right)\right) := \underset{\left(\alpha,\xi\right)\in\mathbb{R}^{2n}}{\arg\min}\; L\left(\alpha,\xi,\mu,\nu\right).$$

- Ensuite il faut chercher le *point selle*:

$$\begin{aligned}\left(\mu^*,\nu^*\right) &= \underset{\left(\mu,\nu\right)\in\left(\mathbb{R}^+\right)^{2n}}{\arg\max}\; L\left(\alpha^*\left(\mu,\nu\right),\xi^*\left(\mu,\nu\right),\mu,\nu\right)\\ &= \underset{\left(\mu,\nu\right)\in\left(\mathbb{R}^+\right)^{2n}}{\arg\max}\;\underset{\left(\alpha,\xi\right)\in\mathbb{R}^{2n}}{\min}\; L\left(\alpha,\xi,\mu,\nu\right).\end{aligned}$$

- La solution au problème sous contrainte est alors:

$$\left(\alpha^*\left(\mu^*,\nu^*\right),\xi^*\left(\mu^*,\nu^*\right)\right).$$

# Minimisation en $\alpha$

$L$ est une fonction quadratique en $\boldsymbol{\alpha}$. Pour trouver le minimum, il suffit donc d'annuler les dérivées:

$$\nabla_{\boldsymbol{\alpha}} L = 2\lambda K \boldsymbol{\alpha} - KY\boldsymbol{\mu} = K\left(2\lambda\boldsymbol{\alpha} - Y\boldsymbol{\mu}\right),$$

ou $Y$ est la matrice diagonale $Y_{i,i} = y_i$. On en déduit:

$$\boldsymbol{\alpha} = \frac{Y\boldsymbol{\mu}}{2\lambda} + \epsilon,$$

avec $K\epsilon = 0$. Mais $\epsilon$ n'intervient pas dans $f$ (cf régression MCR), donc on peut choisir par exemple $\epsilon = 0$ d'ou:

$$\alpha_i^*\left(\boldsymbol{\mu}, \boldsymbol{\nu}\right) = \frac{y_i \mu_i}{2\lambda}, \qquad \textit{pour } i = 1, \ldots, n.$$

# Minmisation en $\xi$

$L$ est une fonction linéaire en $\boldsymbol{\xi}$. Son minimum est donc $-\infty$, sauf si la dérivée est nulle:

$$\frac{\partial L}{\partial \xi_i} = \frac{1}{n} - \mu_i - \nu_i = 0.$$

Donc $\min_{\boldsymbol{\alpha},\boldsymbol{\xi}} L(\boldsymbol{\alpha}, \boldsymbol{\xi}, \boldsymbol{\mu}, \boldsymbol{\nu}) > -\infty$ seulement si $\mu_i + \nu_i = 1/n$ pour $i = 1, \ldots, n$. Comme $\mu_i \geq 0$ et $\nu_i \geq 0$, cela est possible ssi

$$0 \leq \mu_i \leq \frac{1}{n},$$

car on peut alors prendre $\nu_i = 1/n - \mu_i$.

# Problème dual

On calcule donc:

$$L\left(\boldsymbol{\alpha}^*\left(\boldsymbol{\mu},\boldsymbol{\nu}\right),\boldsymbol{\xi}^*\left(\boldsymbol{\mu},\boldsymbol{\nu}\right),\boldsymbol{\mu},\boldsymbol{\nu}\right) =$$

$$\begin{cases} -\infty & \text{si } \mu_i < 0 \text{ ou } \mu_i > 1/n, \\ \sum_{i=1}^n \mu_i - \frac{1}{4\lambda}\sum_{i,j=1}^n y_i y_j \mu_i \mu_j K\left(\mathbf{x}_i,\mathbf{x}_j\right) & \text{sinon.} \end{cases}$$

Le problème dual est donc:

$$\max_{0\leq\boldsymbol{\mu}\leq 1/n} \sum_{i=1}^n \mu_i - \frac{1}{4\lambda}\sum_{i,j=1}^n y_i y_j \mu_i \mu_j K\left(\mathbf{x}_i,\mathbf{x}_j\right).$$

# Retour au primal

En replacant $\mu_i$ par $2\lambda\alpha_i y_i$, on obtient une nouvelle expression du problème primal:

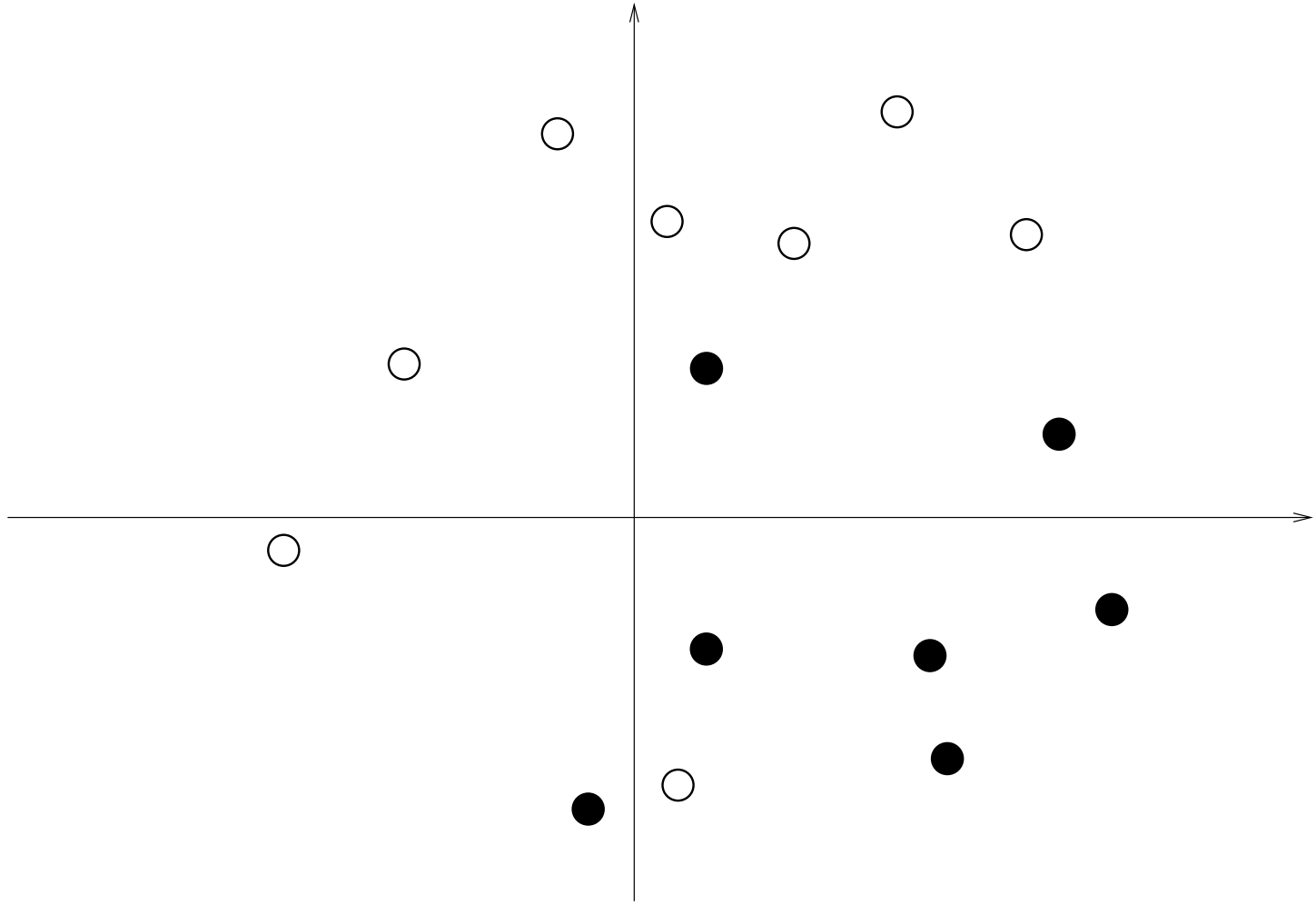
$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^d} 2\sum_{i=1}^n \alpha_i y_i - \sum_{i,j=1}^n \alpha_i\alpha_j K\left(\mathbf{x}_i,\mathbf{x}_j\right)$$

$$= \max_{\boldsymbol{\alpha}\in\mathbb{R}^d} 2\boldsymbol{\alpha}^\top \mathbf{y} - \boldsymbol{\alpha} K \boldsymbol{\alpha},$$

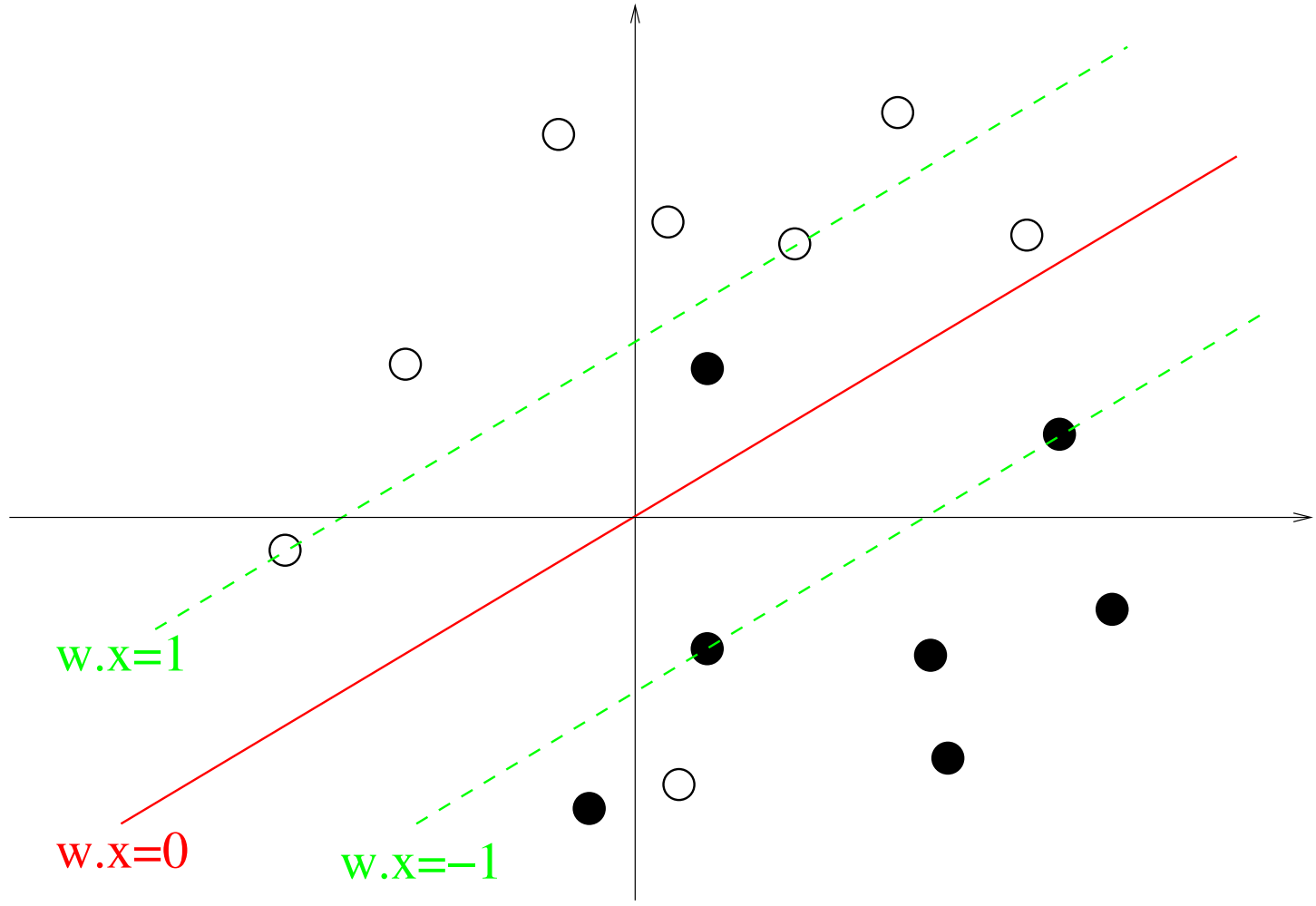
sous la contrainte:

$$0 \leq y_i\alpha_i \leq \frac{1}{2\lambda n}, \qquad \textit{pour } i = 1,\ldots,n.$$

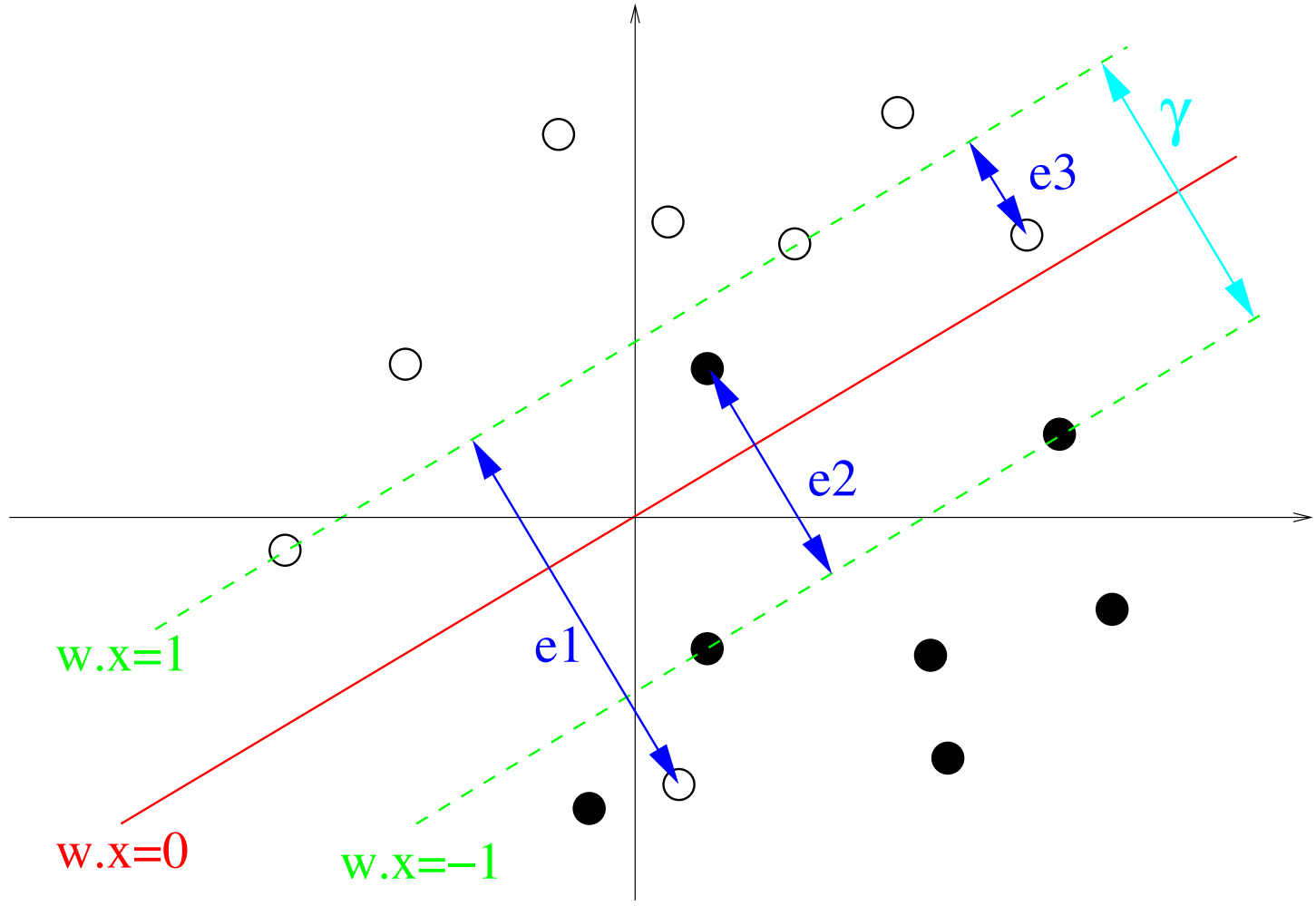
# Interprétation géométrique



# Interprétation géométrique



# Interprétation géométrique



# Interprétation géométrique

Des calculs élémentaires montrent que:

$$\gamma = \frac{2}{\| \mathbf{w} \|} = \frac{2}{\| f_w \|},$$

et

$$V_{hinge}\left(f\left(\mathbf{x}_i\right), y_i\right) = \frac{2e_i}{\gamma}$$

donc la SVM est solution de:

$$\min_{\mathbf{w} \in \mathbb{R}^d} \frac{\sum_{i=1}^n e_i}{\gamma n} + \frac{\lambda}{\gamma^2}.$$

# Conditions de KKT

On rappelle (Kusch-Kuhn-Tucker) que pour la solution du problème d'optimisation sous contraintes, on a, pour $i = 1, \ldots, n$:

$$\begin{cases} \mu_i \left[ y_i f\left(\mathbf{x}_i\right) + \xi_i - 1 \right] = 0, \\ \nu_i \xi_i = 0, \end{cases}$$

avec

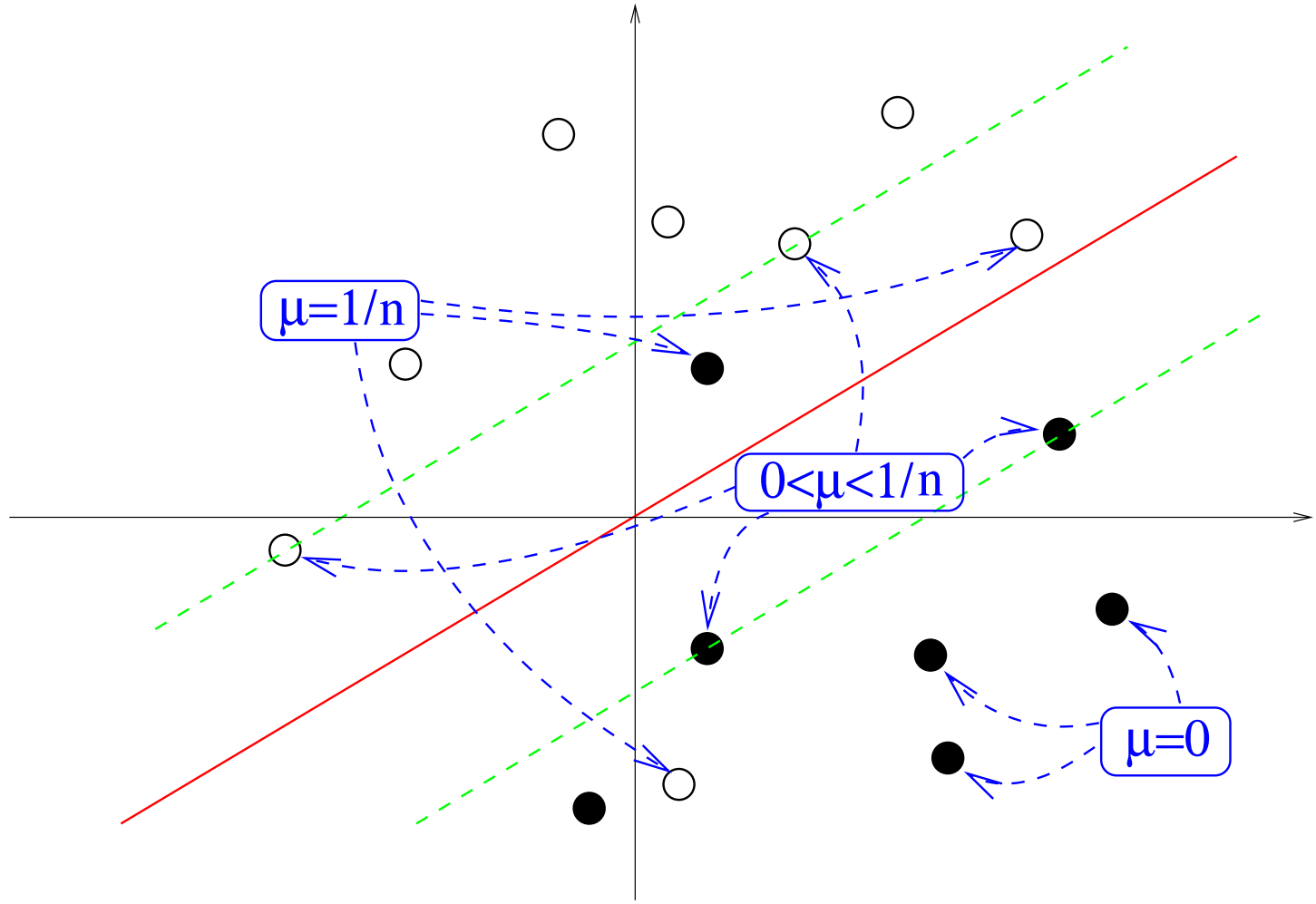
$$\begin{cases} 0 \leq \mu_i \leq \frac{1}{n}, \\ \mu_i + \nu_i = \frac{1}{n}. \end{cases}$$

# Interprétation de $\mu$

- Si $\mu_i = 0$, alors $\nu_i = 1/n > 0$ donc $\xi_i = 0$: $y_i f(\mathbf{x}_i) \geq 1$
- Si $0 < \mu_i < 1/n$ alors $0 < \nu_i$, donc les deux contraintes sont actives: $\xi_i = 0$ et $y_i f(\mathbf{x}_i) + \xi_i - 1 = 0$. On en déduit $y_i f(\mathbf{x}_i) = 1$
- Si $\mu_i = 1/n > 0$ alors $\nu_i = 0$, donc $\xi_i \geq 0$ et $y_i f(\mathbf{x}_i) \leq 1$

Les points avec $\mu_i > 0$ sont appelés les *vecteurs de support*.

# Interprétation de $\mu$



# Vecteurs de support

- Les vecteurs de support sont les points pour lesquels $\mu_i > 0$, et donc $\alpha_i > 0$
- La fonction de discrimination est:

$$\forall \mathbf{x} \in \mathcal{X}, \quad f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i K(\mathbf{x}_i, \mathbf{x}) = \sum_{i \in SV} \alpha_i K(\mathbf{x}_i, \mathbf{x}),$$

ou $SV$ est l'ensemble des indices des vecteurs de support : *seuls les vecteurs de support importent pour la classification d'un nouveau point*.

- Typiquement, il y a peu de SV.

# Implementations

Quelques bonnes implémentations:

- Gist
- SVMLight
- Spider

# Remarque: C-SVM

D'habitude on écrit les SVM comme minimisant:

$$\underset{f \in \mathcal{H}_K}{\arg\min} \quad C \sum_{i=1}^{n} V_{hinge}\left(f\left(\mathbf{x}_i\right), y_i\right) + \| f \|_{\mathcal{H}_K}^2 .$$

C'est bien sur équivalent en prenant $C = 1/n\lambda$. On parle alors de *$C$-SVM*. $C$ est souvent un parametre des implémentations.

# Extension 1 (exercice)

Plutot que des fonctions linéaires, on peut chercher des fonctions *affines* de la forme:

$$f_{\mathbf{w},b}\left(\mathbf{x}\right) = \mathbf{w} \cdot \mathbf{x} + b.$$

Dériver la forme duale quand on veut optimiser:

$$\underset{(\mathbf{x},b)\in\mathbb{R}^{d+1}}{\arg\min}\, C \sum_{i=1}^{n} V_{hinge}\left(f_{\mathbf{w},b}\left(\mathbf{x}_i\right), y_i\right) + \| w \|^2.$$

# Extension 2 (exercice)

Dans le cas des fonctions affines de la forme de la forme:

$$f_{\mathbf{w},b}\left(\mathbf{x}\right) = \mathbf{w} \cdot \mathbf{x} + b,$$

une autre version des SVM consiste à minimiser:

$$\underset{(\mathbf{x},b)\in\mathbb{R}^{d+1}}{\arg\min}\, C \sum_{i=1}^{n} V_{hinge}\left(f_{\mathbf{w},b}\left(\mathbf{x}_i\right), y_i\right)^{2} + \| w \|^2.$$

Ecrire la forme duale.