# Machines à Vecteurs Support
# Séparateurs à Vaste Marge
# SVM

Loïc Barrault

Loic.barrault@lium.univ-lemans.fr

# Sources

- La majeure partie provient de : « Fouille de données dans les corpus de textes, Classification supervisée : SVM. » Michèle Jardino, LIMSI

- SVM, Support Vector Machines, Marti Hearst, Berkeley, http://www.sims.berkeley.edu/courses/is290-2/f04/sched.html

  "Using very large corpora/Spelling correction/clustering"

- SVM, Séparateurs à Vastes Marges, Antoine Cornuéjols, Orsay http://www.lri.fr/~antoine

# Plan

- Classification binaire
  - généralités
    - exemples et définition
    - linéaire/non-linéaire
    - séparable/non séparable
  - Perceptron
  - Séparateurs (classifieurs) à Vastes Marges
  - Fonctions noyau

# GÉNÉRALITÉS

# Classification binaire : exemples

- Filtrage du courrier électronique : (spam / non spam)
- Classification des messages (urgent / non urgent)
- Recherche d'information (correct / incorrect)
- Classification des opinions (positive / négative)

- Classifications multiples
  - Transformation en classification binaire
  - Pour chaque classe, 1 classe contre toutes les autres

# Classification binaire

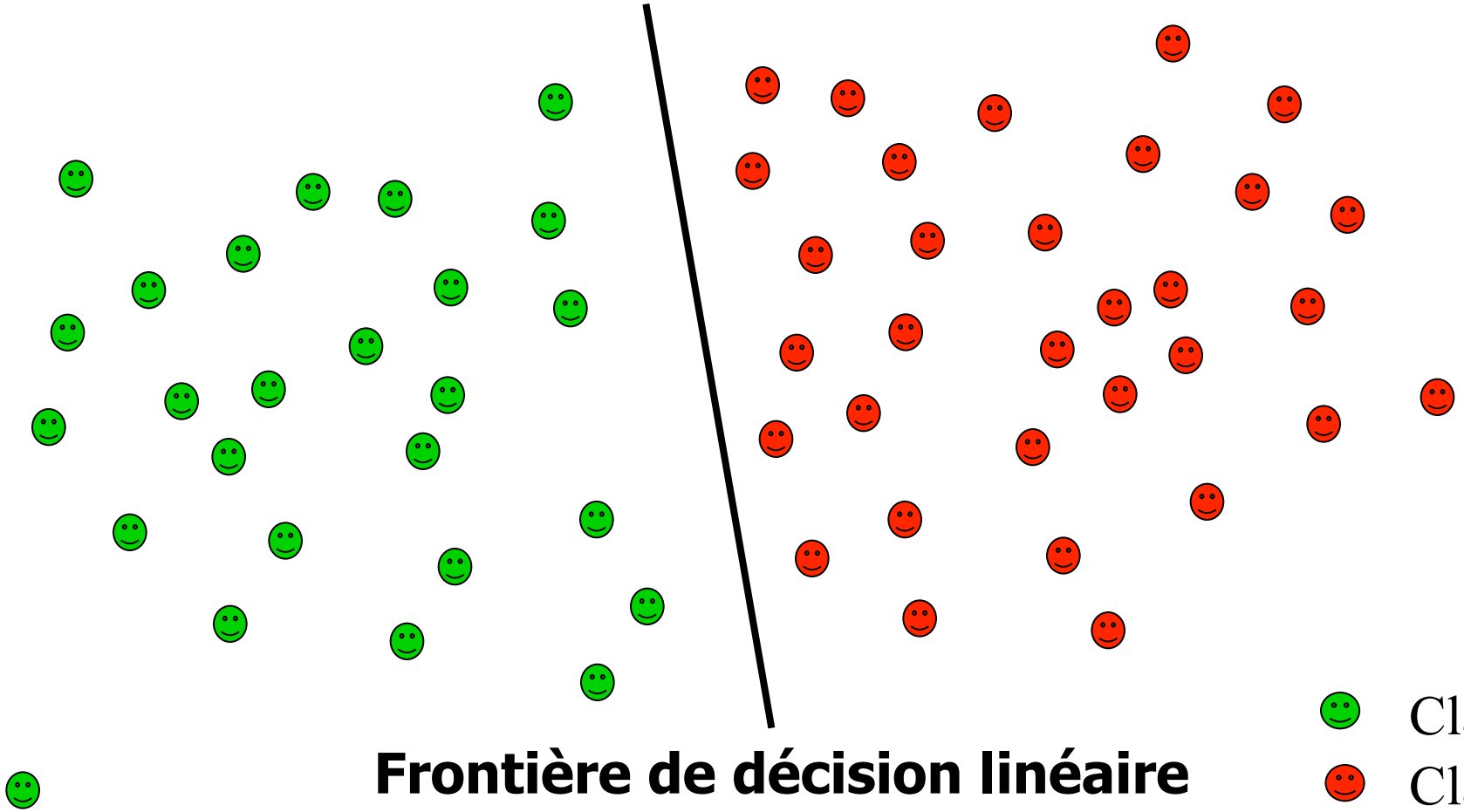
- Données : quelques éléments (textes) qui appartiennent à deux classes différentes
  - classe 1 (+1 ☺ ) et classe 2 (-1 ☹ )

  ou
  - classe positive (+1 ☺ ) et classe négative (-1 ☹ )
- Tâche : entraîner un classifieur sur ces données (dites d'apprentissage) puis prédire la classe d'un nouvel élément (nouveau texte)
- Géométriquement : trouver une séparation entre les deux classes dans l'espace de représentation (*d* dimensions)

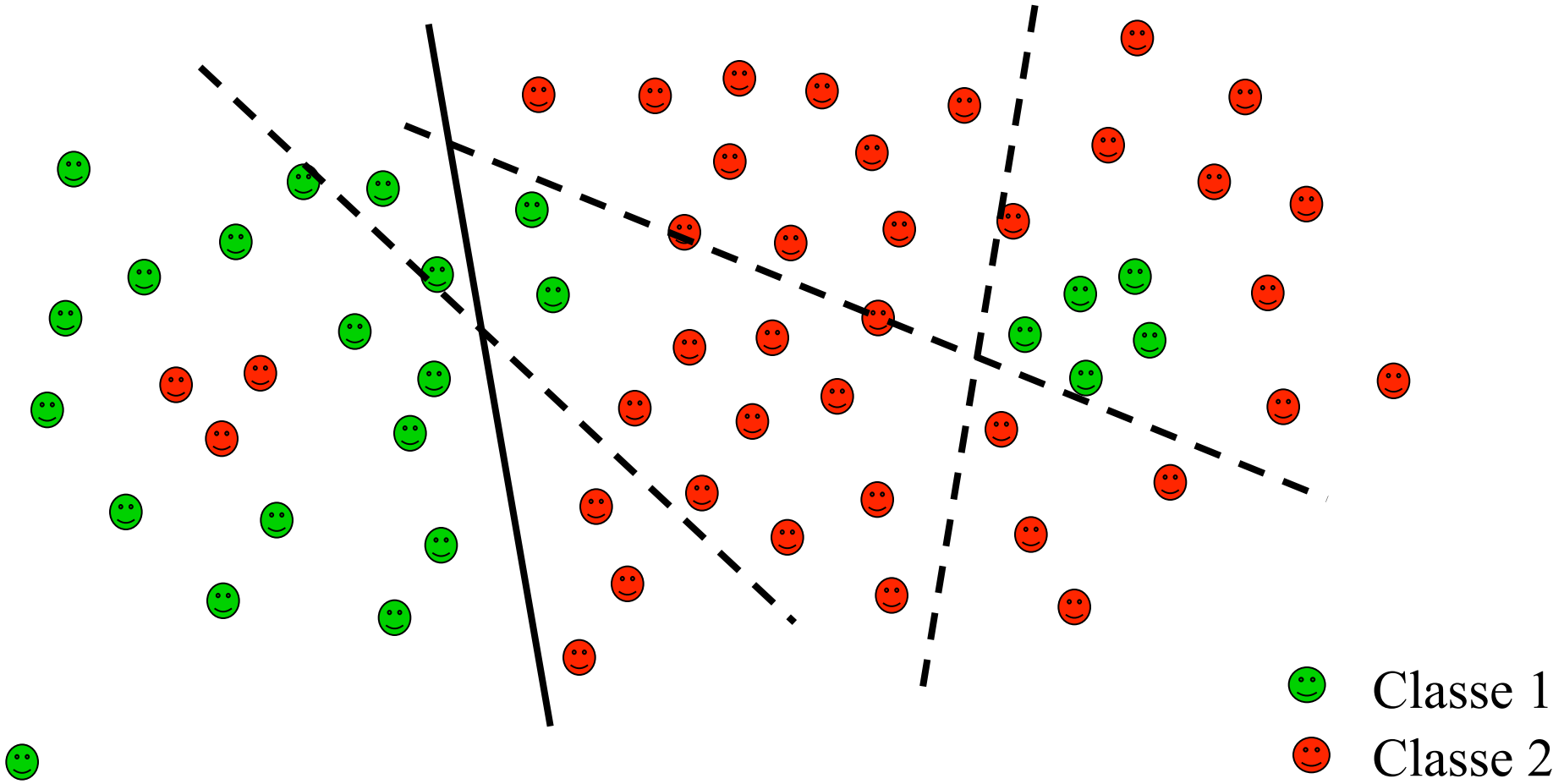
# Séparation linéaire / non linéaire

- **Données  séparables linéairement** :
  - tous les points associés aux données peuvent être séparés correctement par une frontière linéaire
  - hyperplan séparateur
    - Seuil pour un espace de dimension 1
    - Droite pour un espace de dimension 2
    - Plan pour un espace de dimension 3

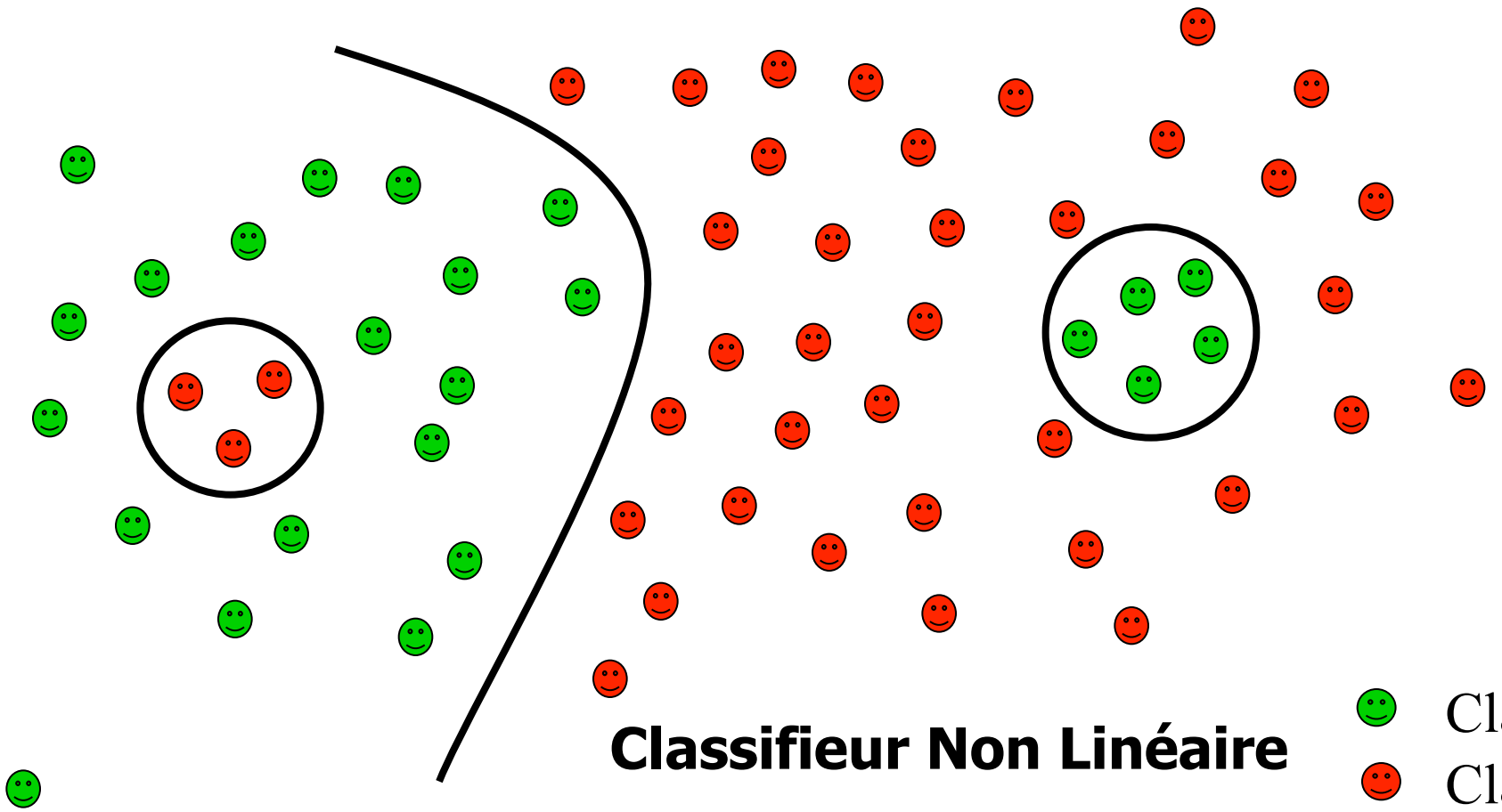
# Données séparables linéairement



# Données non séparables linéairement



# Données non séparables linéairement



# Algorithmes linéaire / non linéaire

- Données séparables linéairement ou non linéairement ?
  - réponse empirique
- Algorithmes Linéaires
  - Algorithmes qui trouvent une frontière linéaire
  - Quand on pense que les données sont linéairement séparables
  - Avantages
    - Simples, peu de paramètres à régler
  - Désavantages
    - Données dans espace de grande dimension sont souvent non linéairement séparables
  - Exemples d'algorithmes : Perceptron, SVM
  - Note : on peut utiliser des  algorithmes linéaires pour des problèmes non linéaires
    - voir fonctions noyau en fin de cours

# Algorithmes linéaire / non linéaire

- Non linéaires
  - Quand les données sont non linéairement séparables
  - Avantages
    - Plus précis
  - Désavantages
    - Plus complexes, plus de paramètres à régler
- Note: la distinction entre linéaire et non linéaire est valable pour la classification multi-classes

# Algorithmes linéaires simples

- Algorithme du Perceptron
  - Réseau de neurones à une couche
  - Linéaire
  - Classification binaire
  - En ligne (apprentissage séquentiel, une donnée à la fois )
  - Apprentissage sur les erreurs

# Algorithmes linéaires simples

- Données : $\{(x_i,y_i)\}_{i=1...n}$
  - x dans $R^d$ (x est un vecteur dans un espace de dimension d)
  - → vecteur de caractéristiques
  - y dans {-1,+1}
  - → étiquette de la classe
- Question:
  - Trouver une frontière linéaire d'équation **wx + b = 0** (hyperplan) telle que la règle de classification associée donne une probabilité d'erreur minimale
  - **règle de classification (décision)**:
    - **y = signe (w x + b) qui signifie** :
    - si wx + b > 0 alors y = +1
    - si wx + b < 0 alors y = -1

From Gert Lanckriet, Statistical Learning Theory Tutorial

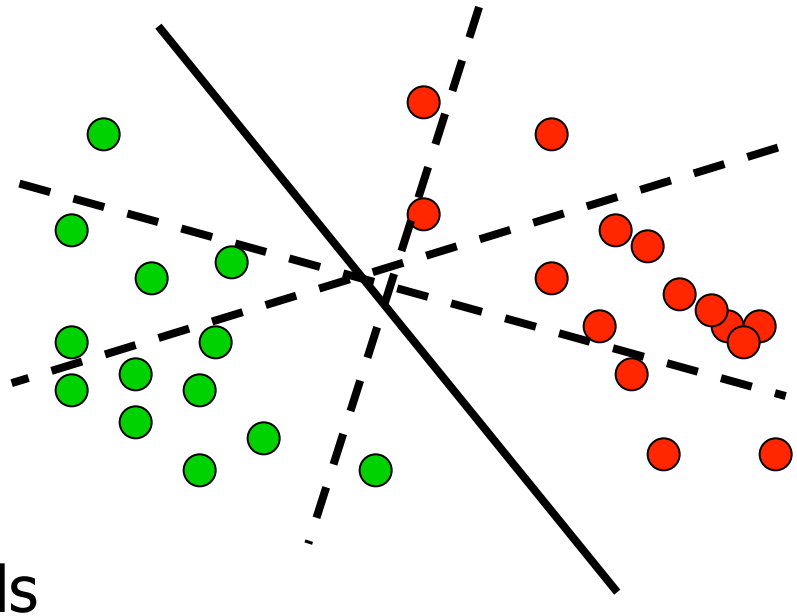
# Classification binaire linéaire

- Trouver un hyperplan

  **(w,b) dans $R^{d+1}$**

  qui classe aussi bien que possible les données (points)

- Progressivement : un point à la fois, en modifiant les poids si nécessaire



**wx + b = 0**

**Règle de Classification :**
**y = signe(wx + b)**

# PERCEPTRON

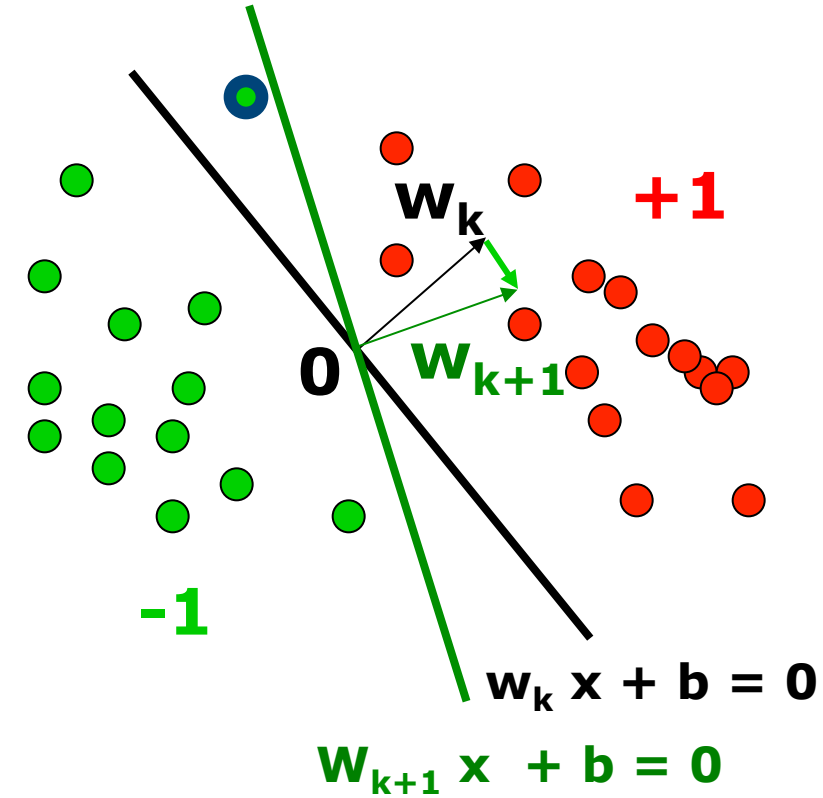
# Algorithme du Perceptron

- Initialisation : $w_1 = 0$
- Mise à jour des poids  Pour chaque point x

```
si classe(x) != decision(x,w)
alors
        w_{k+1} = w_k + y_i x_i
        k  = k + 1
sinon
        w_{k+1} = w_k
```

- **decision(x, w):**

```
si wx + b >= 0 alors renvoie +1
Sinon renvoie -1
```




From Gert Lanckriet, Statistical Learning Theory Tutorial

# Algorithme du Perceptron

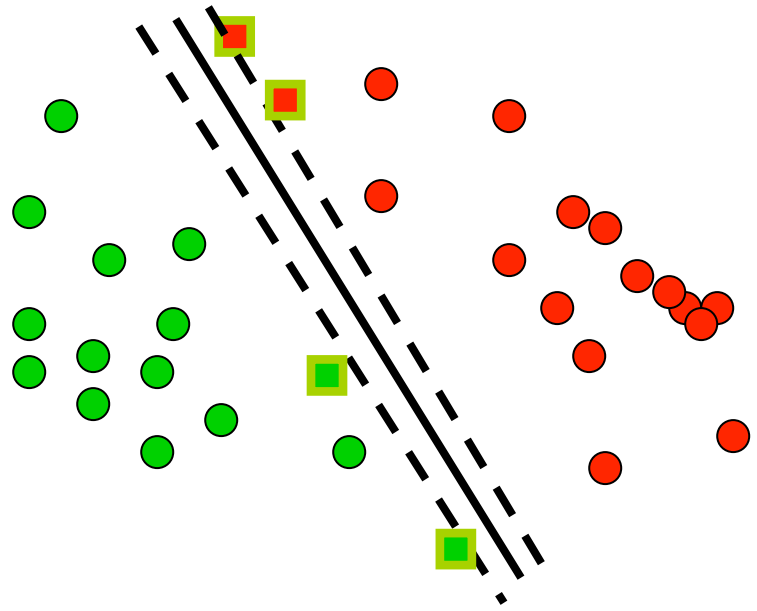
- Progressif : s'adapte toujours aux nouvelles données
- Avantages
  - Simple et efficace
  - Garantie d'apprendre un problème linéairement séparable (convergence, optimum global)
- Limitations
  - Seulement séparations linéaires
  - Converge seulement pour données séparables
  - Pas très efficace dès qu'il y a trop de caractéristiques (**d** trop grand)

From Gert Lanckriet, Statistical Learning Theory Tutorial

- SUPPORT VECTOR MACHINE
- SÉPARATEUR À VASTE MARGE
- MACHINE À VECTEURS SUPPORT

# Séparateur à Vaste Marge

- **Une autre famille d'algorithmes linéaires**
- **Intuition** (Vapnik, 1965)
- Si les classes sont **linéairement séparables** :
  - Séparer les données
  - Hyper-plan “loin” des données :
    - **large marge**
  - résultats statistiques garantis
    - **bonne généralisation**

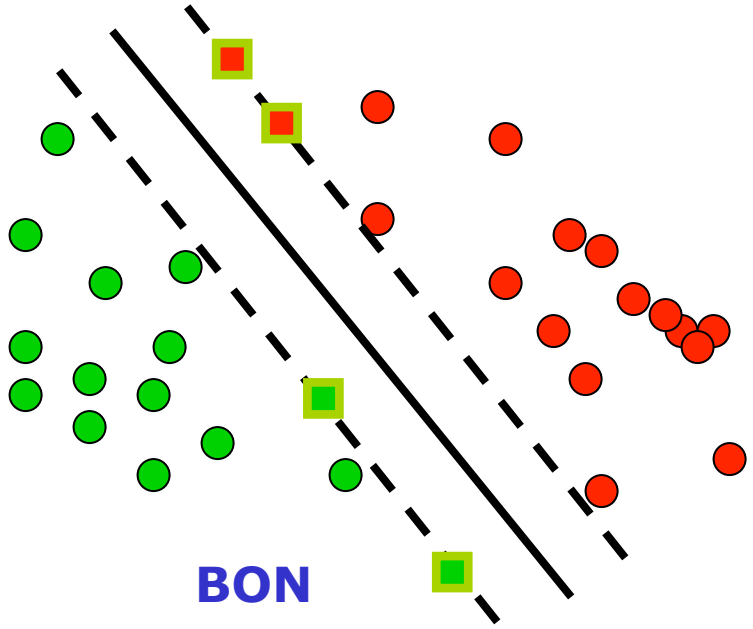


**MAUVAIS**

From Gert Lanckriet, Statistical Learning Theory Tutorial

# Séparateur à Vaste Marge

- **Une autre famille d'algorithmes linéaires**
- **Intuition** (Vapnik, 1965)
- Si les classes sont linéairement séparables :
  - Séparer les données
  - Hyper-plan “loin” des données :
    - **large marge**
  - résultats statistiques garantis
    - **bonne généralisation**



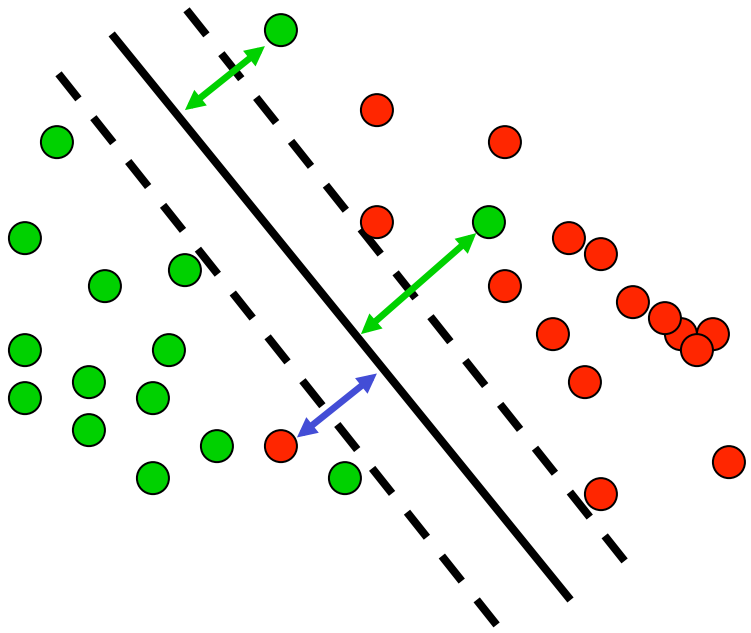

**→ Classifieur à Marge Maximale**

From Gert Lanckriet, Statistical Learning Theory Tutorial

# Séparateur à Vaste Marge

Si **non séparable linéairement**

- **Permettre** quelques **erreurs : perméabilité**
- Essayer encore de placer un  hyperplan “loin” de chaque classe



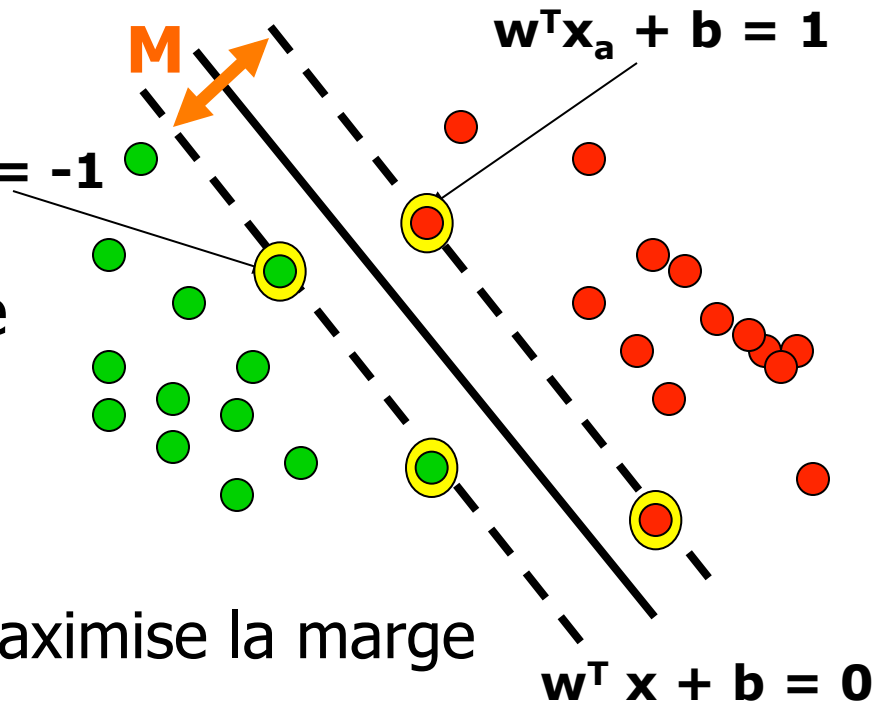
From Gert Lanckriet, Statistical Learning Theory Tutorial

# Séparateur à Vaste Marge

- Avantages
  - Meilleur théoriquement
  - barres d'erreurs mieux connues
- Limitations
  - Calculs plus coûteux
  - Programmation/Optimisation quadratique

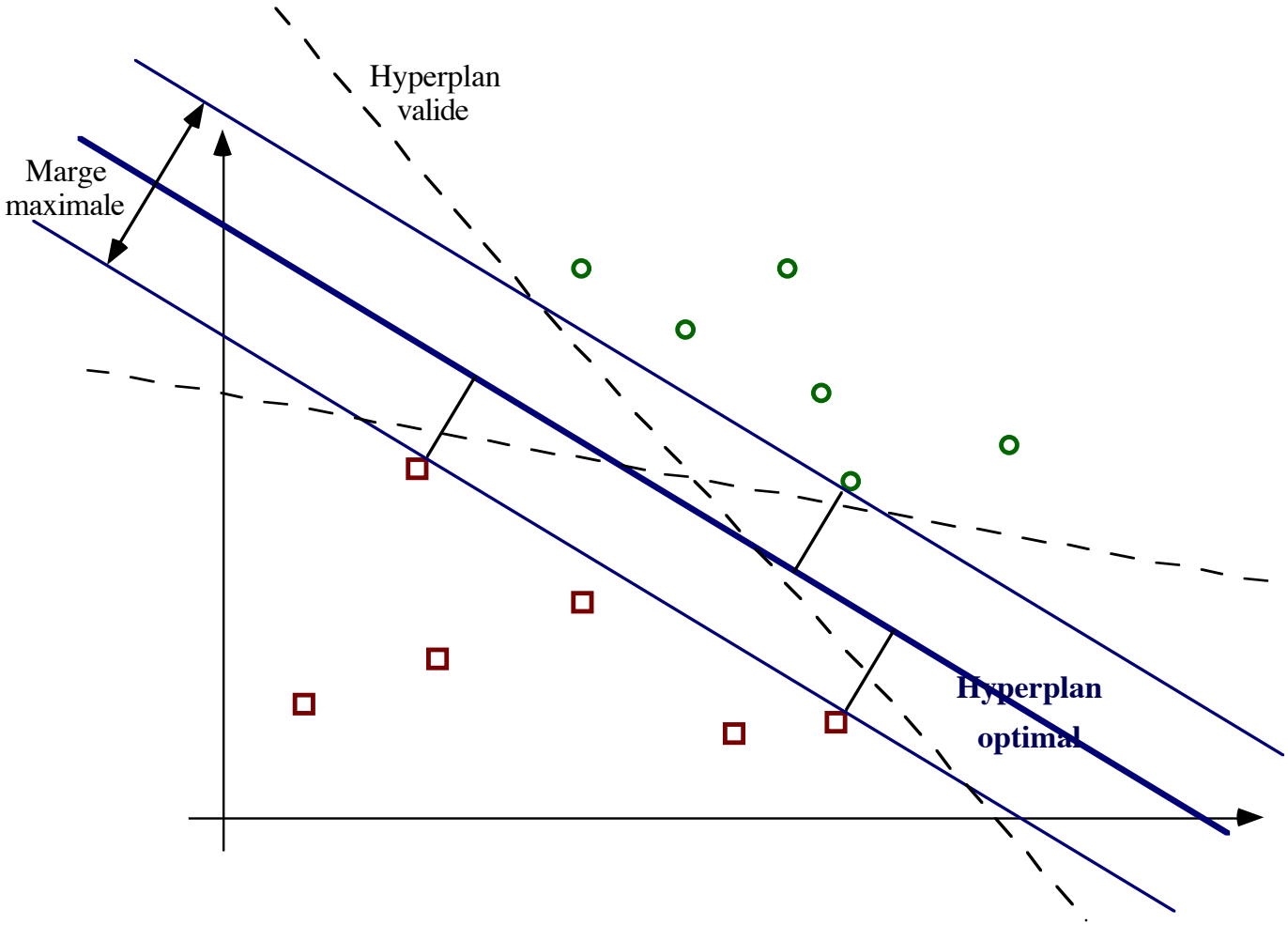
# Vecteurs Support

- Classifieur Vaste Marge
- Cas linéairement séparable
- But :
  trouver l' hyperplan  qui maximise la marge

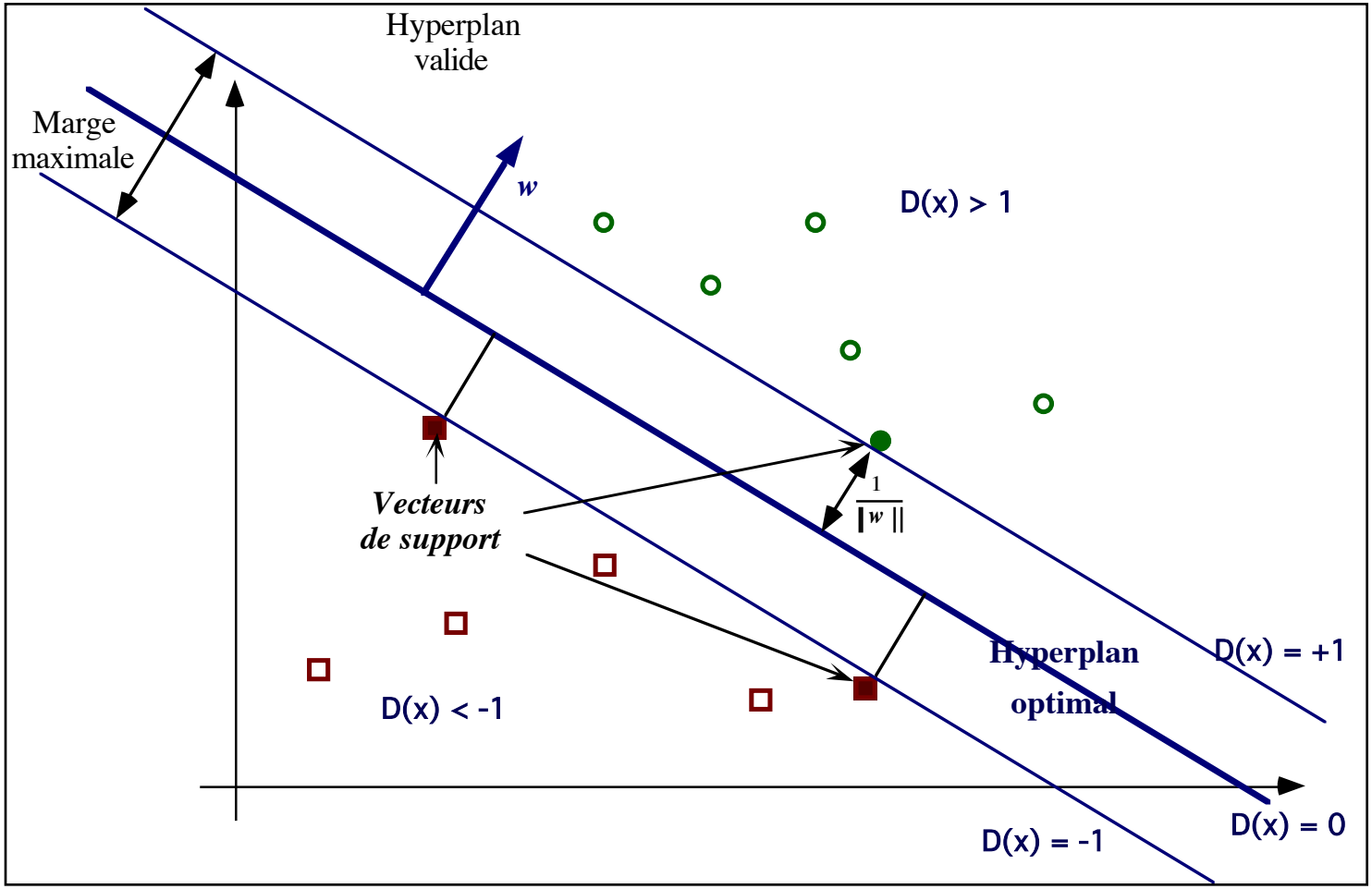



Vecteurs Support

# Hyperplan de plus vaste marge



# Optimisation de la marge



# Optimisation de la marge

- La distance d'un point à l'hyperplan est : $D(\boldsymbol{x}) = \frac{|\boldsymbol{w}.\boldsymbol{x} + b|}{\|\boldsymbol{w}\|}$
- L'hyperplan optimal est celui pour lequel la distance aux points les plus proches est maximale.
- La marge entre les deux classes vaut $\frac{2}{\|\boldsymbol{w}\|}$
- Maximiser la marge revient donc à minimiser $\|\boldsymbol{w}\|$ sous contraintes:

$$\begin{cases} \min \frac{1}{2}\|\boldsymbol{w}\|^2 \\ \forall \boldsymbol{i} \quad y_i(\boldsymbol{w}.\boldsymbol{x}_i + b) \geq 1 \end{cases}$$

# SVMs : un problème d'optimisation quadratique

- Il faut donc déterminer *w* et *b* minimisant :

$$\frac{1}{2}\|\boldsymbol{w}\|^2$$

**EXPRESSION PRIMALE**

(afin de maximiser le pouvoir de généralisation)

- sous les contraintes (hyperplan séparateur) :

$$y_i\left[(\boldsymbol{w}.\boldsymbol{x}_i)+b\right]\geq 1\,,\qquad i=1,\ldots,n$$

# Résolution de la forme primaire du problème


Il faut régler $d$ + 1 paramètres

- Possible quand $d$ est assez petit

  avec des méthodes d'optimisation quadratique

- Impossible quand $d$ est grand (> qqs $10^3$)

# Transformation du problème d'optimisation

- Méthode des multiplicateurs de Lagrange

$$\begin{cases} L(\boldsymbol{w},b,\alpha) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} \alpha_i \{(\boldsymbol{w}.\boldsymbol{x}_i + b)\boldsymbol{y}_i - 1\} \\ \forall i \;\; \alpha_i \geq 0 \end{cases}$$

**EXPRESSION DUALE**

- Problème dual

$$\begin{cases} \max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (\boldsymbol{x}_i . \boldsymbol{x}_j) \\ \forall i \;\; \alpha_i \geq 0 \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases}$$

# Propriétés de la forme duale

- La complexité du problème d'optimisation est
  - $\propto$ n (taille de l'échantillon d'apprentissage)
  - et non $\propto d$ ( taille de l'espace d'entrée )

➥ **Possible d'obtenir des solutions pour des problèmes impliquant $\approx 10^5$ exemples**

# Solution du problème d'optimisation

$$\begin{cases} D(\boldsymbol{x}) = (\boldsymbol{w}^* . \boldsymbol{x} + b^*) \\ \boldsymbol{w}^* = \sum_{i=1}^{m} \alpha_i^* y_i \boldsymbol{x}_i \\ w_0^* = y_s - \sum_{i=1}^{m} \alpha_i^* y_i (\boldsymbol{x}_i . \boldsymbol{x}_s) \end{cases}$$

* : estimé

$(x_S, y_S)$ étant n'importe quel point de support

**Propriété 1 :** seuls les $\alpha_i$ des points les plus proches sont non-nuls : *points de support* ou *vecteurs support* *(exemples critiques)*.

**Propriété 2** : seuls interviennent les produits scalaires entre les observations *x* dans le problème d'optimisation.

# PROBLÈME NON LINÉAIRE

# Problèmes non linéairement séparables dans $\mathcal{X}$

La majorité des problèmes !!!

Idée :

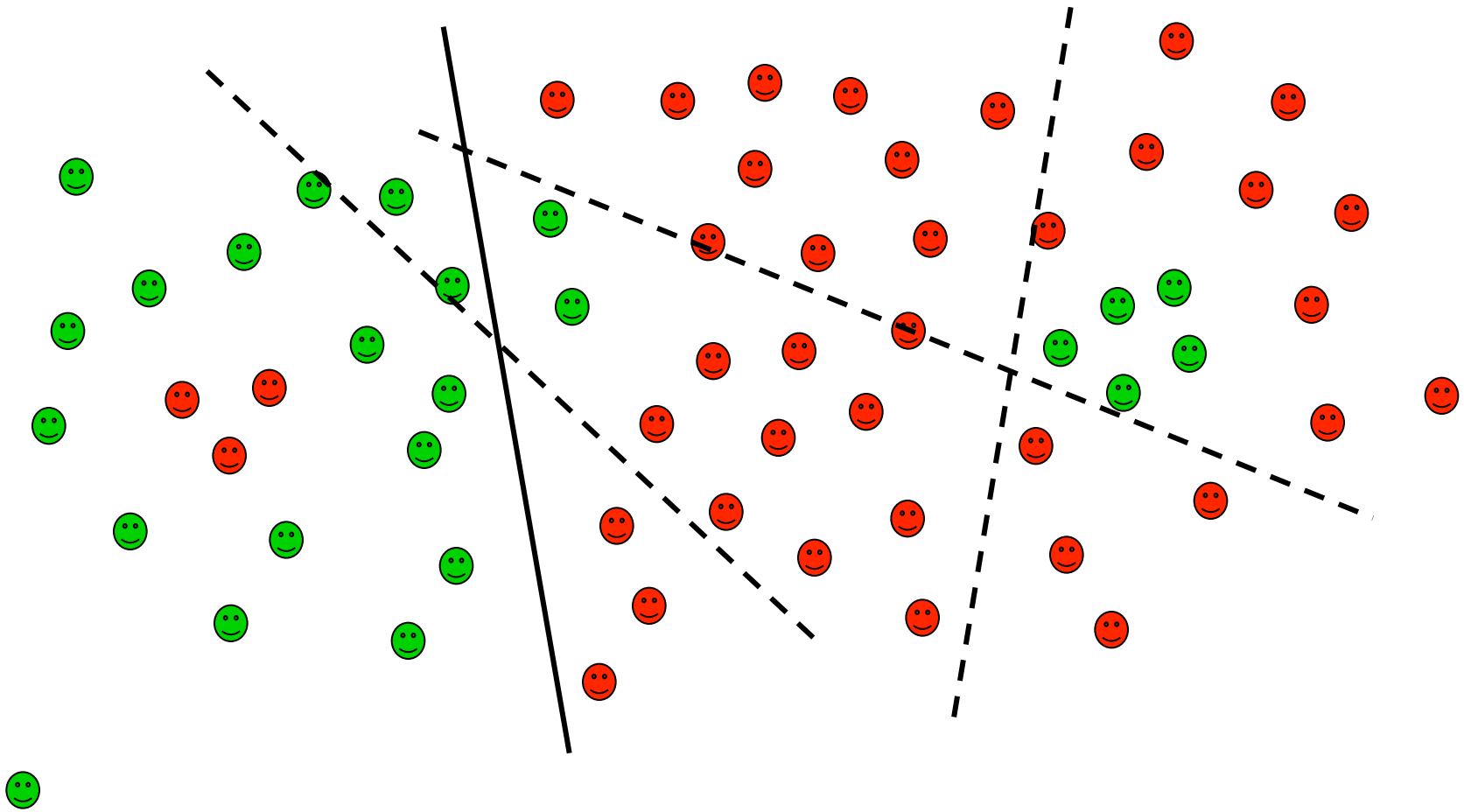
**Projeter dans un espace de redescription de très grande dimension**

➥ Presque toujours le problème devient linéairement séparable

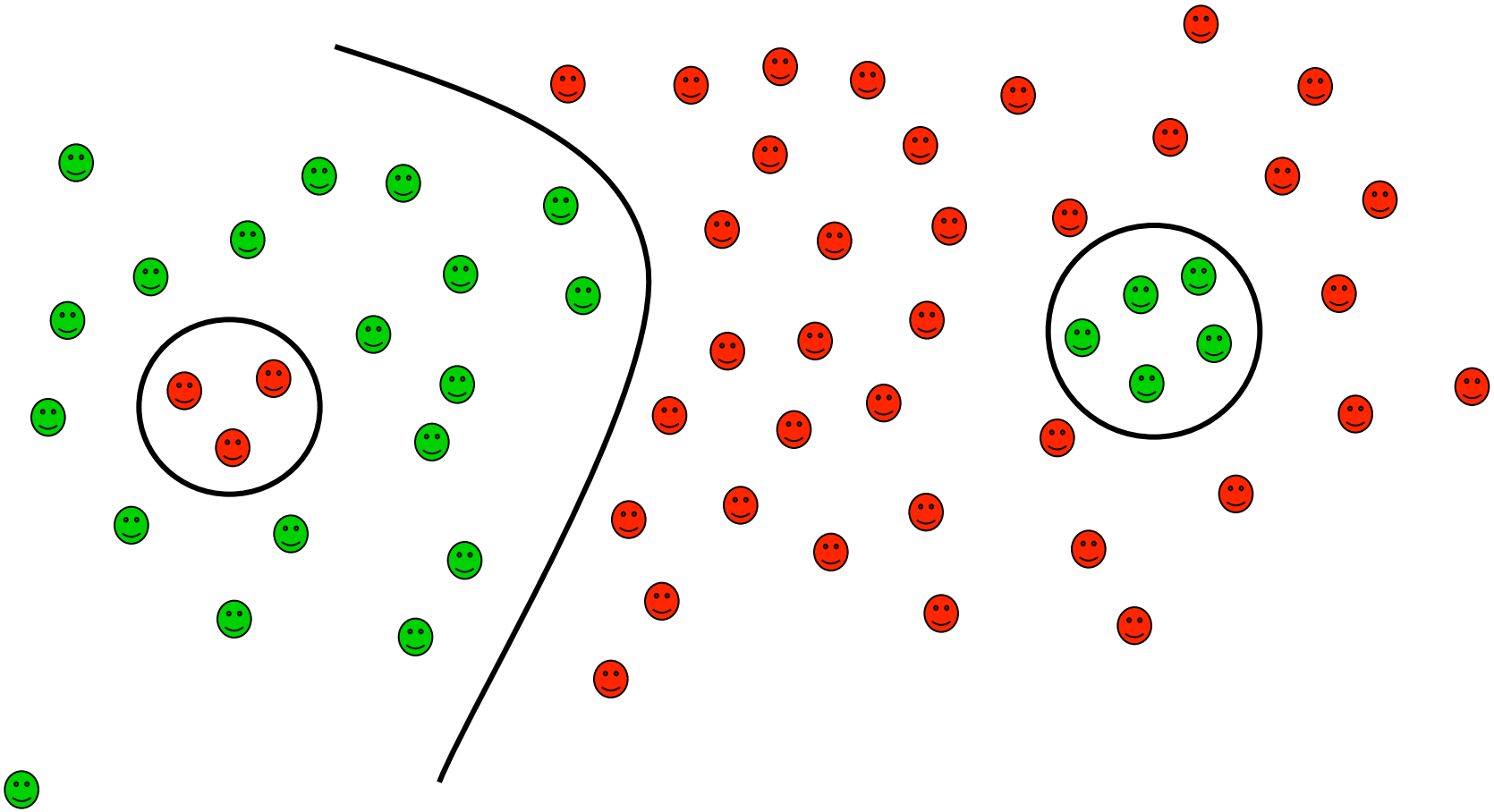
Mais :

- Fléau de la dimensionalité
- *d* explose !!?

# Problème non linéaire



# Problème non linéaire

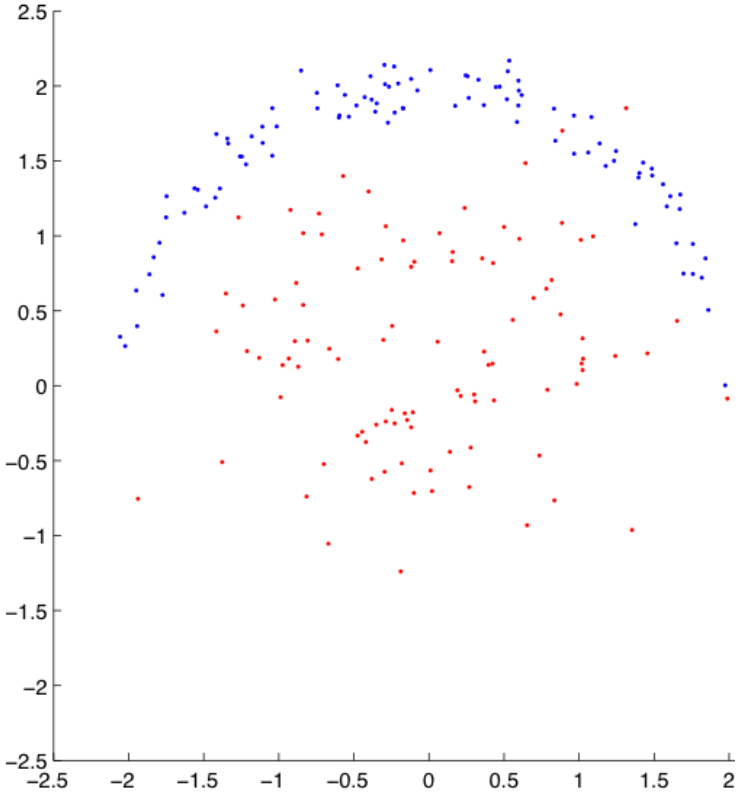


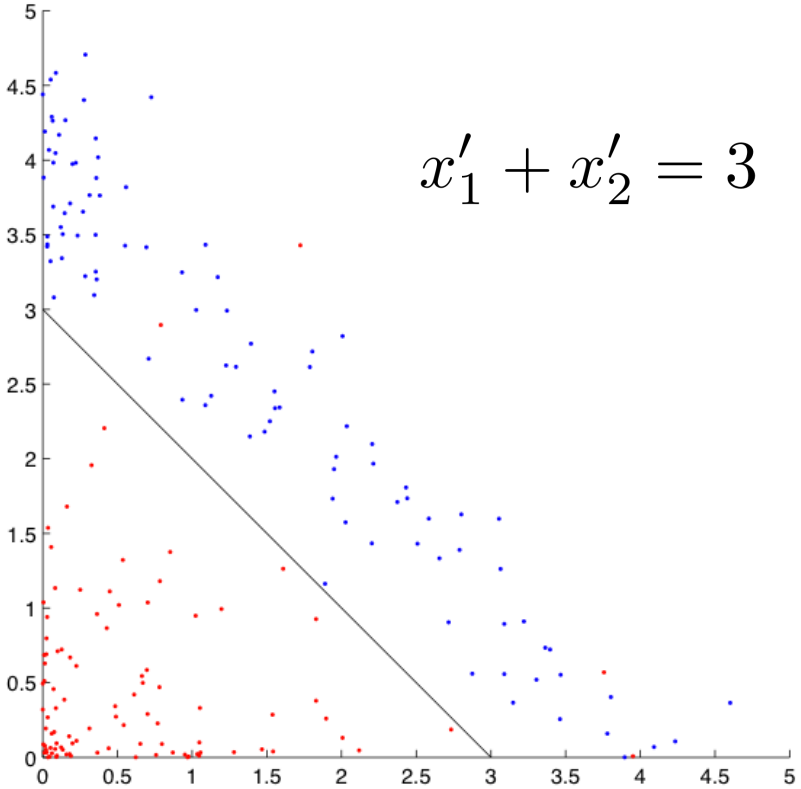
# Fonctions noyau

- Famille d'**algorithmes non linéaires**

- Transforme un problème non linéaire en un problème linéaire
  - Projection des données dans un espace de traits caractéristiques différents
    - de plus grande dimension
  - Utilisation d'algorithmes linéaires dans le nouvel espace

# Fonction noyau



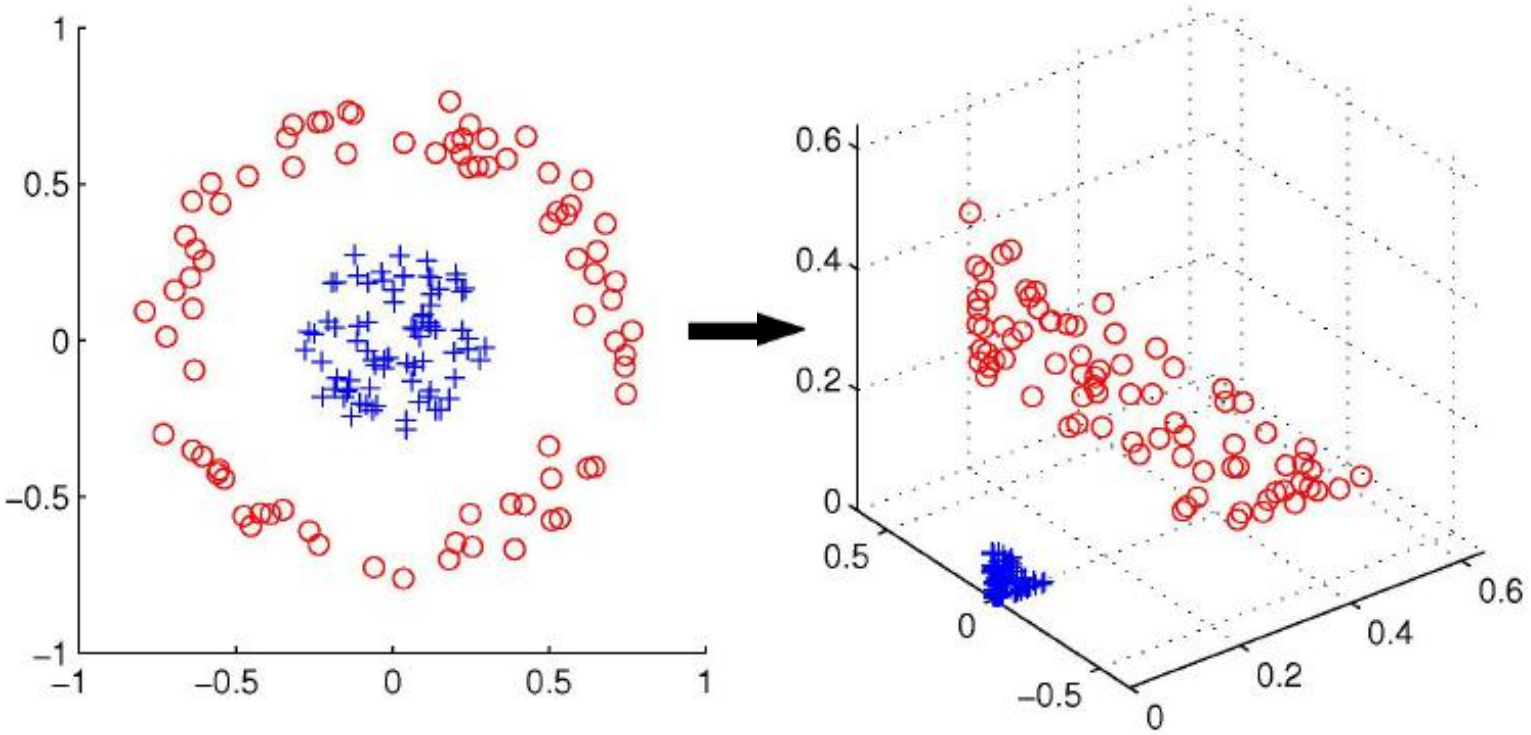




$$x'_1 + x'_2 = 3$$

$$(x_1, x_2) \rightarrow (x_1^2, x_2^2)$$

Issu de A. Cornuéjols

# Fonction noyau

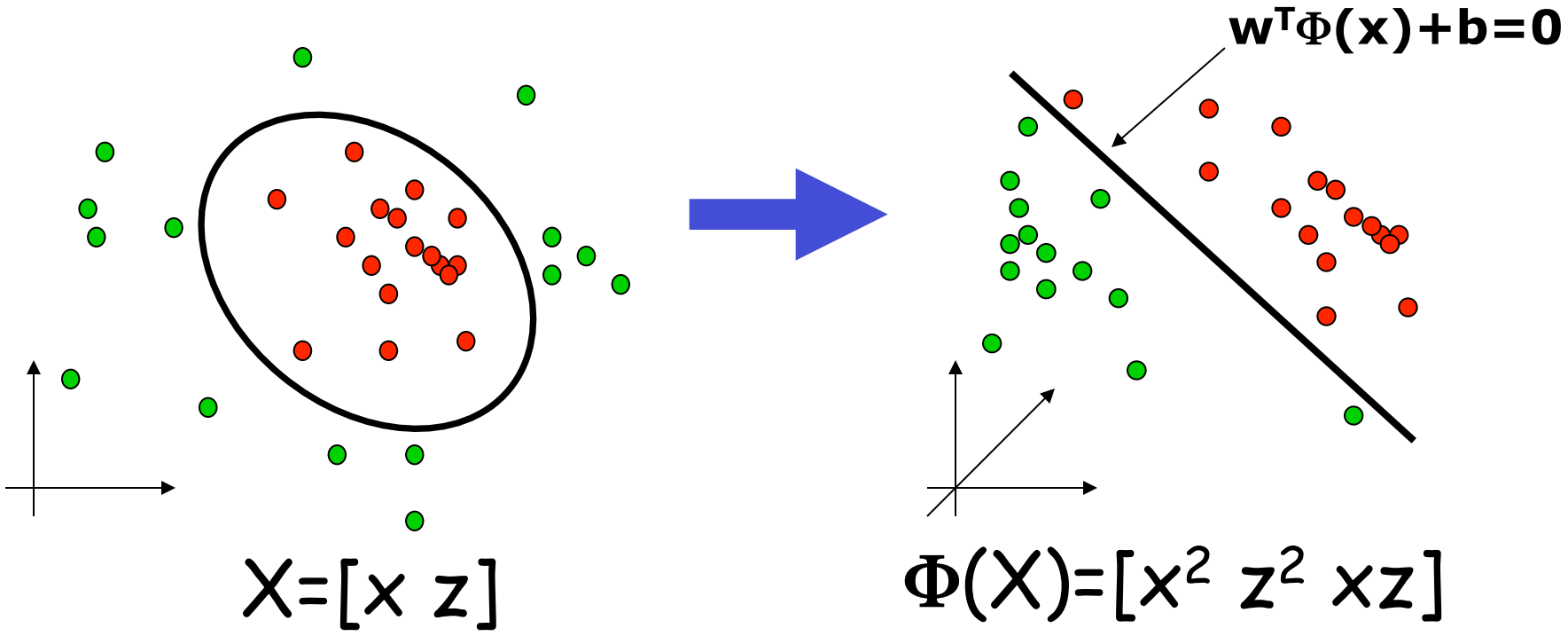



$(x_1, x_2)$ $(x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$

Issu de A. Cornuéjols

# Principe de méthodes à base de fonctions noyau

$\Phi : R^d \rightarrow R^D \quad (D >> d)$




$$f(x) = \text{signe}(w_1x^2+w_2z^2+w_3xz +b)$$

From Gert Lanckriet, Statistical Learning Theory Tutorial

# Le nouveau problème d'optimisation

- Soit $\Phi$ : $\mathcal{X}$ -> $\Phi(\mathcal{X})$, on peut remplacer partout $x$ par $\Phi(x)$
- Si $\Phi$ est bien choisie, $K(x, x') = \Phi(x).\Phi(x')$ peut être facile à calculer et le problème devient :

$$\begin{cases} \max\limits_{\alpha} \sum\limits_{i=1}^{n} \alpha_i - \frac{1}{2} \sum\limits_{i=1}^{n} \sum\limits_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ \forall i \;\; 0 \le \alpha_i \le C \\ \sum\limits_{i=1}^{n} \alpha_i y_i = 0 \end{cases}$$

# Solution du nouveau problème d'optimisation

- La fonction de décision devient :

$$D(\boldsymbol{x}) = \sum_{j=1}^{n} w_j \, g_j(\boldsymbol{x})$$

$n$ : nb de fonctions de base (peut être très grand)

- Soit dans la forme duale :

$$D(\boldsymbol{x}) = \sum_{i=1}^{m_S} \alpha_i \, y_i \, K(\boldsymbol{x}_i, \boldsymbol{x}) + b$$

$m_S$ : nb de points de support

# Fonctions noyau usuelles (1/2)

- Polynomiale : polynômes de degré *q*
  - **fonction noyau associée :**

$$K(\boldsymbol{x},\boldsymbol{x}') = (\boldsymbol{x}.\boldsymbol{x}' + 1)^q$$

- RBF : fonctions à base radiale
  - **fonction noyau associée :**

$$h(\boldsymbol{x}) = sign\left(\sum_{i=1}^{n} \alpha_i \exp\left\{-\frac{|\boldsymbol{x}-\boldsymbol{x}_i|^2}{\sigma^2}\right\}\right)$$

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

- Sigmoïde : réseaux de neurones
  - **fonction noyau associée :**

$$h(\boldsymbol{x}) = sign\left(\sum_{i=1}^{n} \alpha_i \tanh\left\{v(\boldsymbol{x}.\boldsymbol{x}_i) + a\right\} + b\right)$$

$$K(\boldsymbol{x},\boldsymbol{x}') = \tanh\left(a\boldsymbol{x}.\boldsymbol{x}' - b\right)$$

# Les fonctions noyau

- … encodent :
  - Une **mesure de similarité** sur les données

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

  - Les **fonctions de décision**
  - Le **type de régularisation** réalisée
    - ex : les fonctions gaussiennes favorisent les solutions régulières
  - Le **type de covariance** dans l'espace des entrées
    - ex : fonctions noyau invariantes par rotation
  - Sorte de **distribution de probabilité *a priori*** sur l'espace des hypothèses

# Cas du problème non séparable : marges douces

- On introduit des variables "ressort" qui pénalisent l'erreur commise :

$$\begin{cases} \min \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{l}\xi_i \\ \forall \boldsymbol{i} \;\; y_i(\boldsymbol{w}.\boldsymbol{x}_i + b) \geq 1 - \xi_i \end{cases}$$

- Le problème dual a la même forme à l'exception d'une constante *C*

$$\begin{cases} \max_{\alpha} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j (\boldsymbol{x}_i.\boldsymbol{x}_j) \\ \forall i \;\; 0 \leq \alpha_i \leq C \\ \sum_{i=1}^{n}\alpha_i y_i = 0 \end{cases}$$

C grand : on est laxiste

C petit : on est strict

# RÉALISATIONS

# La mise en pratique

- Il faut choisir :
  - Le **type** de fonction **noyau *K***
    - sa forme
    - ses paramètres
  - La valeur de la constante ***C***

- **La sélection rigoureuse de ces paramètres exige une estimation de la dimension de Vapnik-Chervonenkis et l'application de la borne de généralisation** ε
  - Dans le cas séparable, il est possible de déterminer ces paramètres
  - Dans le cas non séparable, il faut tester avec des méthodes empiriques pour faire le meilleur choix

# Exemple : données d'apprentissage

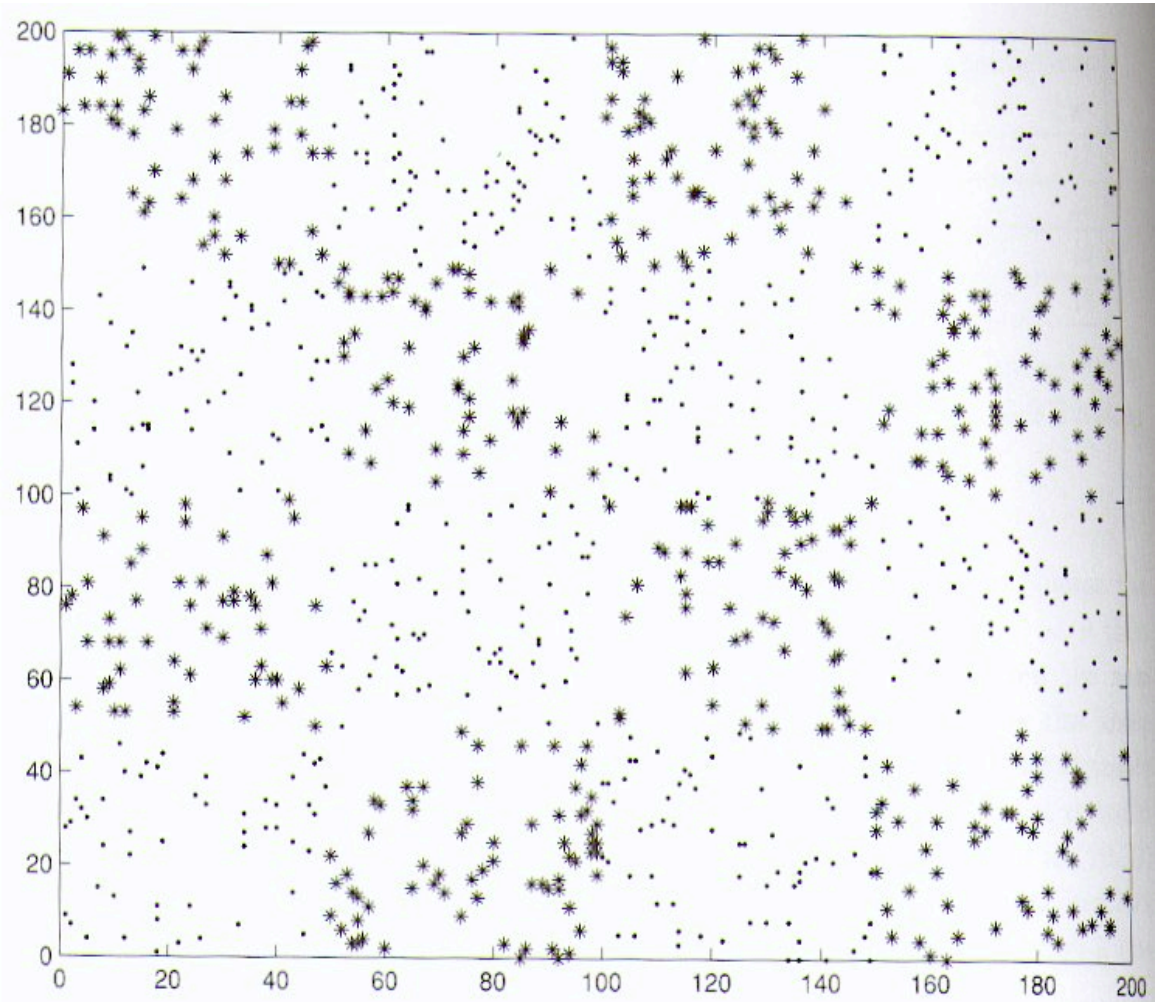
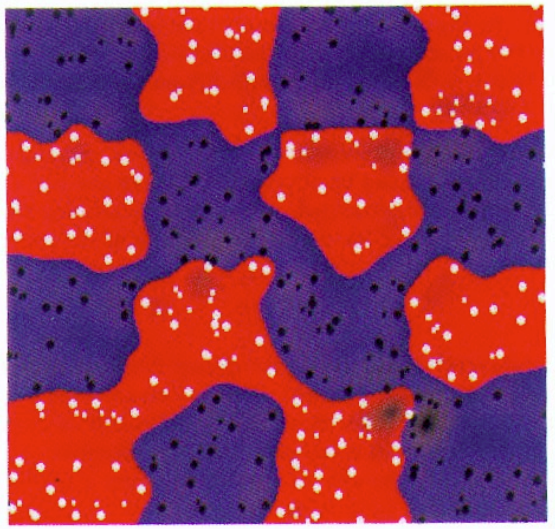



**Figure 10.2** Training points for checkerboard pattern

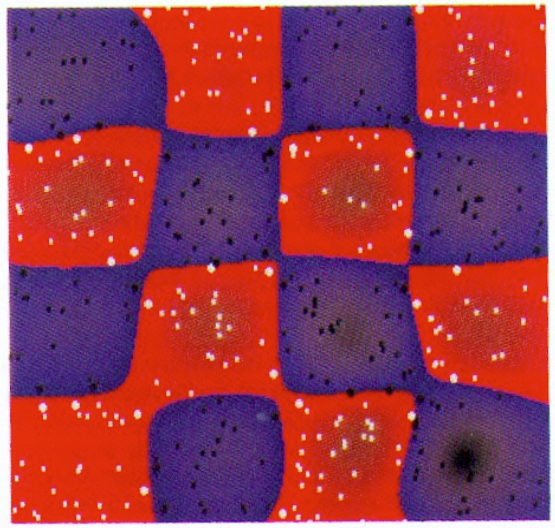
# Effet des paramètres de contrôle

- Apprentissage de deux classes
  - exemples tirés uniformément sur l'échiquier
- SVM à noyau gaussien (base radiale)

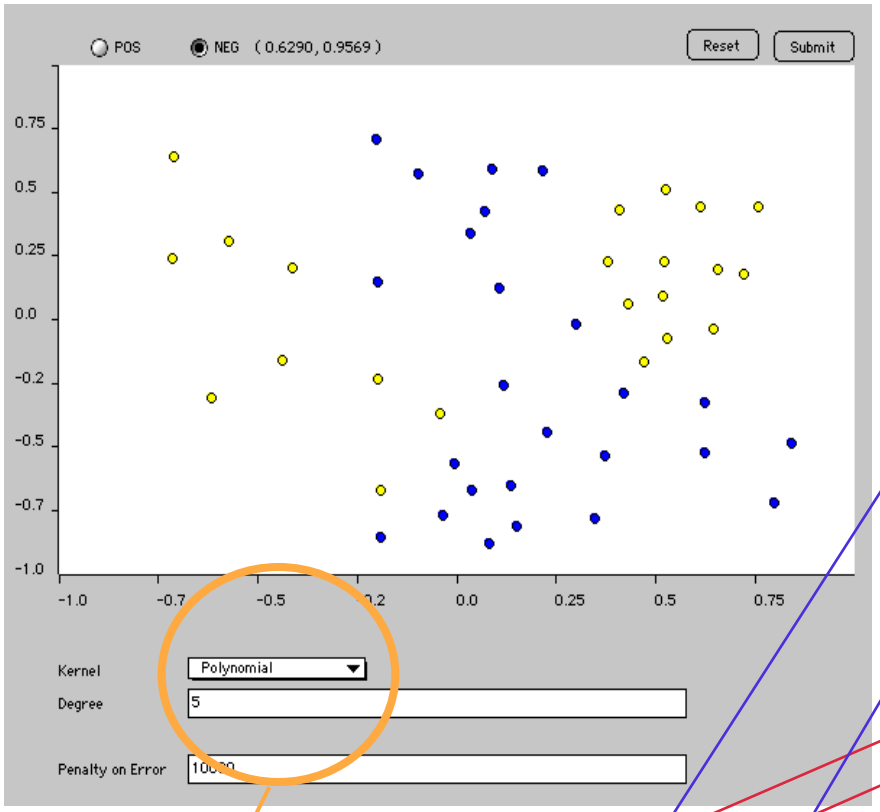
$$K(\boldsymbol{x},\boldsymbol{x}') = e^{-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\sigma^2}}$$

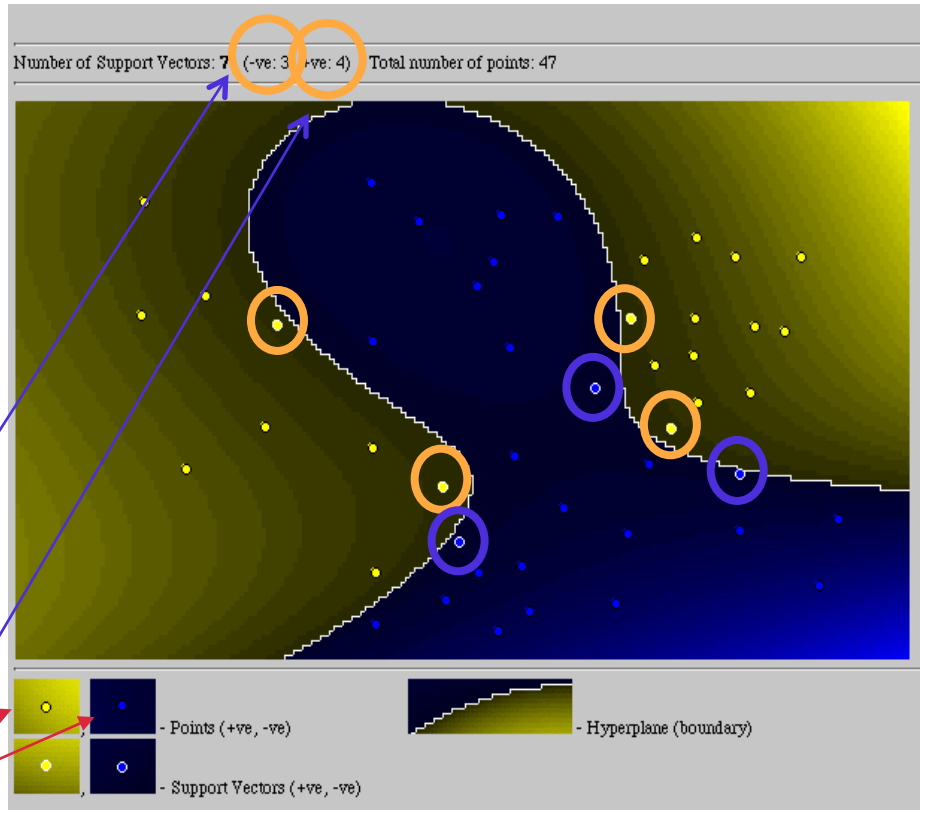


- Ici deux valeurs de $\sigma$
  - En haut : petite valeur
  - En bas : grande valeur
- Les gros points sont des exemples critiques
  - Plus en haut qu'en bas
- Dans les deux cas : $R_{emp} = 0$



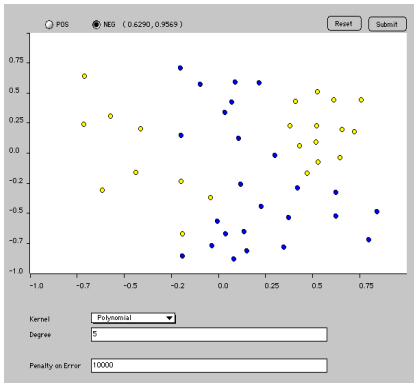
# Paramètres de contrôle : les fonctions noyau

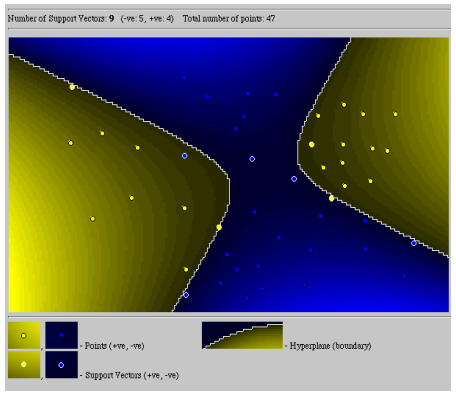


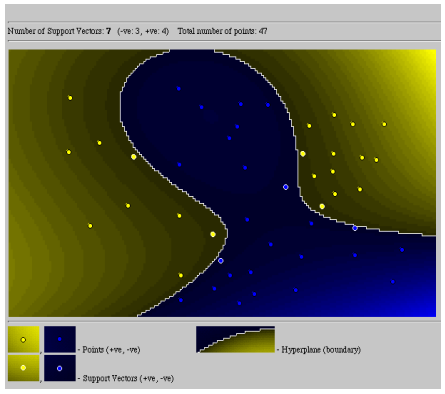



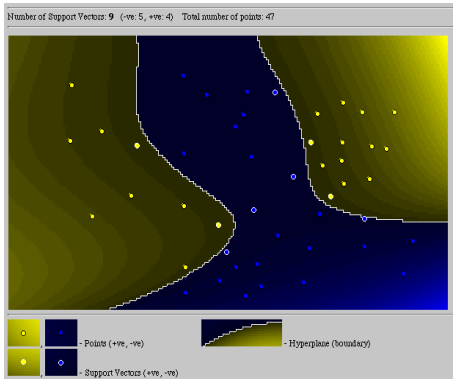

- 47 exemples (22 +, 25 -)
- *Exemples critiques* : 4 + et 3 -
- **Ici** *fonction polynomiale* **de** *degré 5* **et** *C = 10000*
- `http://svm.dcs.rhbnc.ac.uk/pagesnew/GPat.shtml`

# Paramètres de contrôle : fonctions noyau



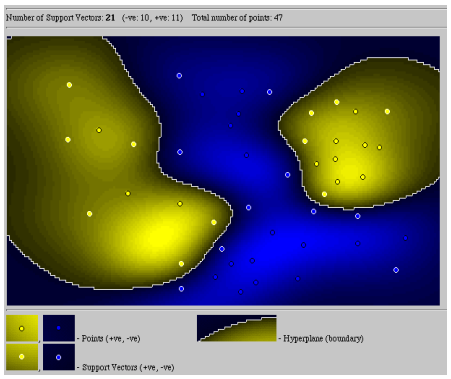



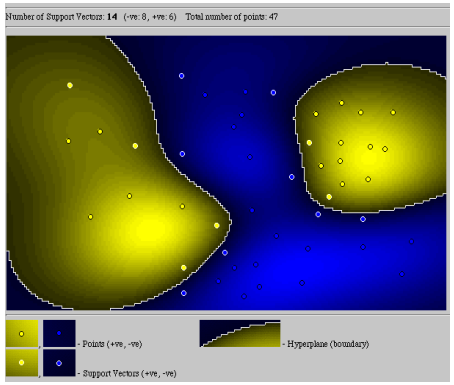



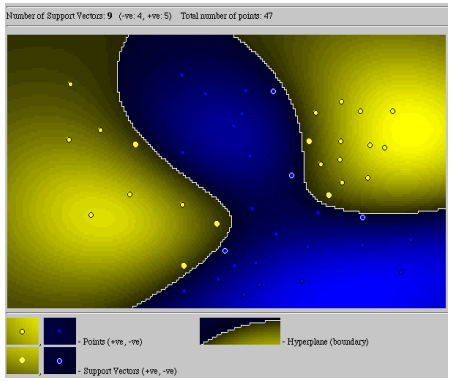

**(5-, 4+)** **(3-, 4+)** **(5-, 4+)**

- 47 exemples (22 +, 25 -)
- *Exemples critiques* : 4 + et 3 -

**Ici** *fonction polynomiale* **de** *degré* $2, 5, 8$ **et** $C = 10000$



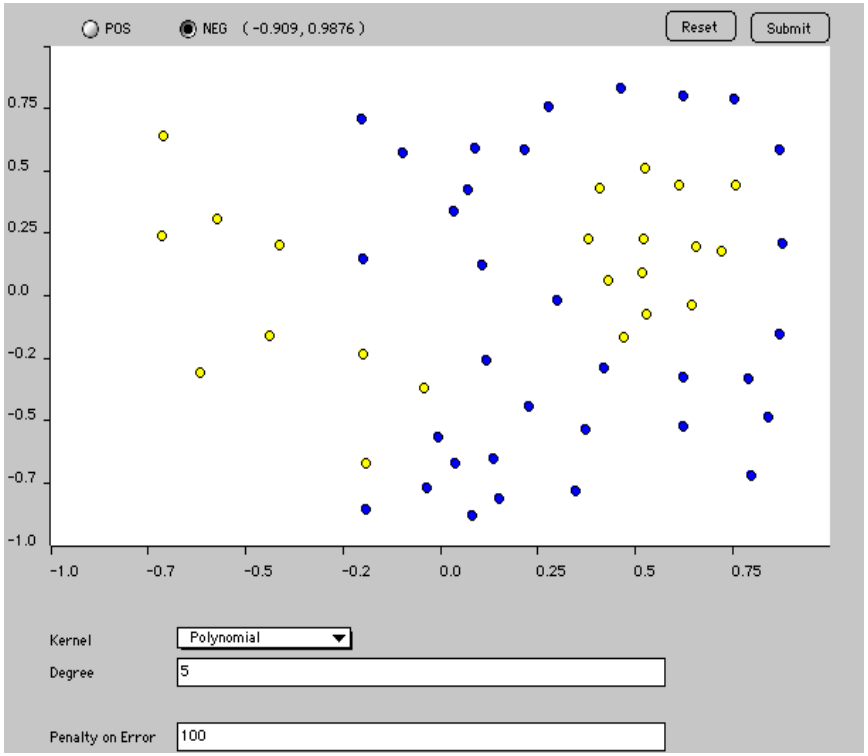



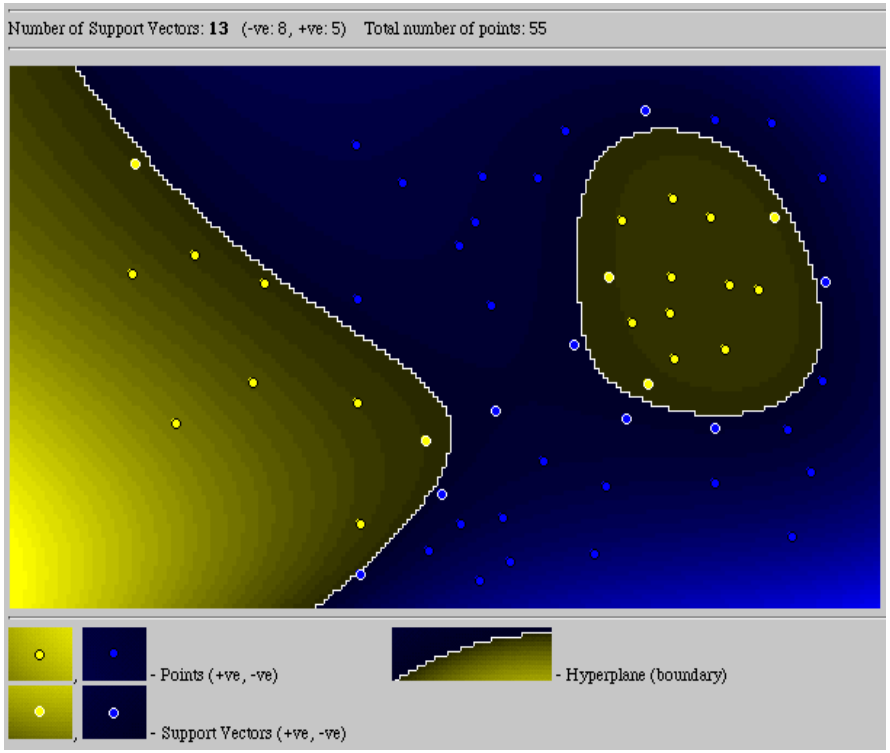




**(10-, 11+)** **(8-, 6+)** **(4-, 5+)**

**Ici** *fonction Gaussienne* **de** $\sigma = 2, 5, 10, 20$ **et** $C = 10000$

# Ajout de quelques points ...




- 47 + 8 exemples (30 +, 25 -)
- *Exemples critiques* : 5 + et 8 -
- **Ici** *fonction polynomiale* **de** *degré 5* **et** *C = 10000*
- `http://svm.dcs.rhbnc.ac.uk/pagesnew/GPat.shtml`

# Domaines d'application des SVMs

- Traitement d'images
  - Reconnaissance de caractères manuscrits
  - Reconnaissance de scènes naturelles
  - Reconnaissance de visages

- *Entrées* : image bidimensionnelle en couleur ou en niveaux de gris
- *Sortie* : classe (chiffre / personne)

# Domaines d'application des SVMs

- Catégorisation de textes
  - Classification d'e-mails
  - Classification de pages web

  - *Entrées* : document texte, html, etc.
    - Approche « sac de mots »
    - Document = vecteur de mots (lemmatisés pondérés par tf-idf)
  - *Sortie* : catégorie (thème, spam/non-spam)
  - ***Noyau*** :
    - Produit scalaire des vecteurs
    - $C = \infty$ (marge dure)

# Domaines d'application des SVMs

- Diagnostic médical
  - Évaluation du risque de cancer
  - Détection d'arythmie cardiaque
  - Évaluation du risque d'accidents cardio-vasculaires à moins de 6 ans

- *Entrées* : état du patient (sexe, age, bilan sanguin, …)
- *Sortie* :
  - Classe : à risque ou non
  - Probabilité d'accident à échéance donnée

# Implémentation des SVMs

- Minimisation de fonctions différentiables convexes à plusieurs variables
  - Pas d'optima locaux
  - Problèmes de stockage de la matrice noyau
    - si milliers d'exemples
    - Long dans ce cas
  - D'où mise au point de méthodes spécifiques
    - Gradient sophistiqué
    - Méthodes itératives, optimisation par morceaux
  - Plusieurs **packages publics** disponibles
    - Weka (utilisation en fouille de données / M2)
    - mySVM [http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/index.html]
    - SVMTorch [http://bengio.abracadoudou.com/SVMTorch.html]
    - SVM$^{Light}$ [http://svmlight.joachims.org/]

# Bilan

- SVMs très utilisés
  - Méthode générale
  - Facile d'emploi
- Résultats **en général** équivalents et souvent meilleurs
- **Stimulent tout un ensemble de travaux sur des** méthodes à base de noyaux **(*kernel-based methods*)**
- Limites
  - **Problèmes i.i.d.** (données indépendantes et identiquement distribuées)

# Sources documentaires

- Ouvrages / articles
  - Cornuéjols & Miclet (02) : *Apprentissage artificiel. Concepts et algorithmes*. Eyrolles, 2002.
  - Cristianini & Shawe-Taylor (00) : *Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.
  - Herbrich (02) : *Learning kernel classifiers*. MIT Press, 2002.
  - Schölkopf, Burges & Smola (eds) (98) : *Advances in Kernel Methods : Support Vector Learning*. MIT Press, 1998.
  - Schölkopf & Smola (02) : *Learning with kernels*. MIT Press, 2002.
  - Smola, Bartlett, Schölkopf & Schuurmans (00) : *Advances in large margin classifiers*. MIT Press, 2000.
  - Vapnik (95) : *The nature of statistical learning.* Springer-Verlag, 1995.
- Sites web
  - `http://www.kernel-machines.org/` **(point d'entrée)**
  - `http://www.support-vector.net` **(point d'entrée)**

# Pourquoi ça marche ?

## La marge est liée à la capacité en généralisation

- **Normalement, la classe des hyperplans de $\mathbb{R}^d$ est de $d_{\mathcal{H}} = d + 1$**
- Mais la classe des hyperplans de marge $\frac{1}{\|w\|}$ tq. $\|w\|^2 \leq c$ est bornée par : $d_{\mathcal{H}} \leq \min(R^2 c, d) + 1$

**où $R$ est le rayon de la plus petite sphère englobant l'échantillon d'apprentissage $S$**

➥ Peut être beaucoup plus petit que la dimension $d$ de l'espace d'entrée $\mathcal{X}$

# Exemples à voir sur :

Démo :

http://svm.research.bell-labs.com/

http://svm.dcs.rhbnc.ac.uk/pagesnew/GPat.shtml

http://cs.stanford.edu/people/karpathy/convnetjs/

# Domaines d'application des SVMs

- Étude de séquences en bio-informatique
  - Biologie structurale prédictive (prédiction de structure secondaire du génome)
  - Identification de régions codantes de l'ADN génomique
  - Phylogénie …

- *Entrées* : chaînes d'acides aminées
- *Sortie* :
  - Structure secondaire
  - Intron / exon
  - Ancêtre
- ***Noyau*** relationnel :
  - Modèle génératif
    (chaînes de Markov : insertion, délétion, remplacement, …)