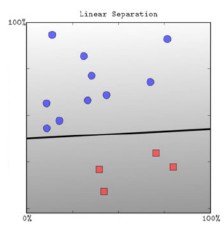
SPLEX
Statistiques pour la classification et fouille de données en génomique
Perceptrons, méthodes à noyaux

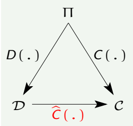
Pierre-Henri WUILLEMIN

DEcision, Système Intelligent et Recherche opérationnelle
LIP6
pierre-henri.wuillemin@lip6.fr
http://webia.lip6.fr/~phw/splex

Rappel : classification linéaire binaire



Définition (CLB)

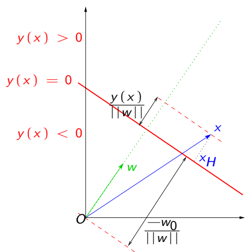


- $\mathcal{C} = \{\ominus, \oplus\}$
- $\exists w \in \mathbb{R}^d, w_0 \in \mathbb{R}, \exists f : \mathbb{R} \to \mathcal{C},$

$$\forall x \in \mathbb{R}^d, \widehat{C}(x) = f\left(\sum_{i=1}^{d} w_i \cdot x_i + w_0\right)$$

Le problème d'apprentissage : trouver w, w_0 et f.

$C(x) = f(y(\mathbf{x}))$ avec
- $y(\mathbf{x}) = \sum_{i=1}^{d} w_i \cdot x_i + w_0$
- $y(\mathbf{x}) = \mathbf{w}' \cdot \mathbf{x} + w_0$
- $y(\mathbf{x}) = \mathbf{w}^{+'} \cdot \mathbf{x}^{+}$ (voir page suivante).



Séparabilité

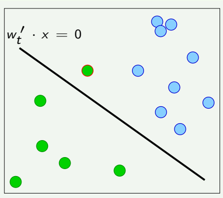
$$\forall j \in \{1, \cdots, N\}, C(\mathbf{x}_j) \cdot y(\mathbf{x}_j) > 0$$

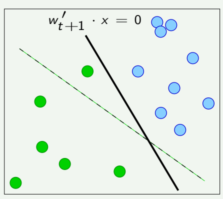
Perceptron linéaire

notation : En notant $w = \begin{bmatrix} w_0 \\ w_1 \\ \dots \\ w_d \end{bmatrix}$ et $x = \begin{bmatrix} 1 \\ x_1 \\ \dots \\ x_d \end{bmatrix}$, $\sum_{i=1}^{d} w_i \cdot x_i + w_0 = w' \cdot x$.

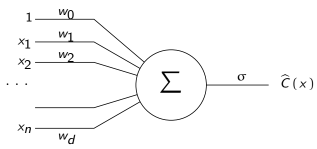
Apprentissage d'un perceptron linéaire

Data : $\{\mathbf{x}_i, C(\mathbf{x}_i)\}_{i=1,\dots,N}$
Initialisation de w : w_1;
$t = 1$;
repeat
  Tirer aléatoirement un exemple : $\mathbf{x}_i$;
  if $C(\mathbf{x}_i)(w_t' \cdot \mathbf{x}_i) \geq 0$ **then**
    $w_{t+1} \leftarrow w_t$;
  else
    $w_{t+1} \leftarrow w_t + \epsilon C(\mathbf{x}_i)\mathbf{x}_i$;
  $t = t + 1$;
until *(critère d'arrêt satisfait)*;





Synthèse : Perceptron linéaire



Avantages

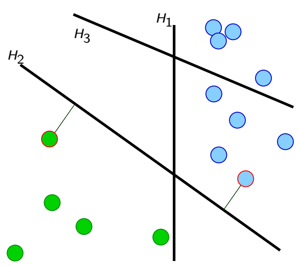
- Algorithme incrémental (adaptation aux nouvelles données)
- Algorithme simple
- Si données séparables alors garantie de convergence et de maximum global

Inconvénients

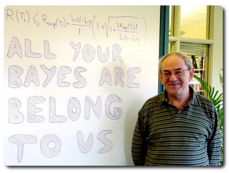
- Non-unicité de la solution (et non-déterminisme de l'algorithme)
- Convergence lente (quand d augmente)
- Si données non séparables : pas de convergence, pas de terminaison de l'algorithme.

Optimisation de l'hyper-plan séparateur

Lorsque les données sont séparables, il n'y a pas unicité de l'hyper-plan séparateur.



- H_3 ne sépare pas.
- H_1 et H_2 séparent.
- H_2 meilleur car plus grande marge.



Principes

- chercher le CLB de marge maximum (cf. Vapnik, 1965)
- Les points les plus proches (définissant la marge) sont appelés :
  vecteurs supports ou exemples critiques
- Ce classifieur minimise la *valeur de la marge d'erreur probable maximum* du CLB (cf. *dimension de Vapnik-Chervonenkis*).

Programme d'optimisation de la marge (1)

- Supposons que l'hyperplan $(w' \cdot x + w_0 = 0)$ sépare correctement les données Π_a. On sait alors que :

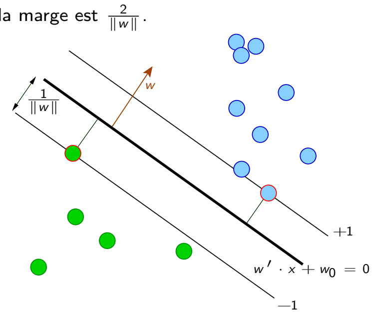
$$\forall x \in D_a, C(x)\left(w' \cdot x + w_0\right) > 0$$

- Donc, $\exists B, \forall x \in D_a, C(x)\left(w' \cdot x + w_0\right) \geq B$.
- Il suffit de multiplier w par un scalaire pour augmenter arbitrairement B. Il faut donc plus contraindre w :
  pour tout vecteur support x, $\left|w' \cdot x + w_0\right| = 1$
- Or la distance de x à l'hyperplan est $\frac{|w' \cdot x + w_0|}{\|w\|}$. Donc la taille de la marge est $\frac{2}{\|w\|}$.

D'où le programme d'optimisation de (w, w_0) :

Maximisation de la marge

$$\left| \begin{array}{l} \max_w \frac{2}{\|w\|} \text{ ou } \min_w \frac{1}{2}\|w\|^2 \\ \\ \forall x \in D_a, C(x)\left(w' \cdot x + w_0\right) \geq 1 \end{array} \right.$$



⚠ programmation quadratique (résolvable si d assez petit.)

Si $d >>$: Fonction de coût et contraintes convexes (Th. de Kuhn-Tucker) $\Rightarrow$ Forme duale !

Optimisation de la marge (2) : Lagrangien

Lagrangien

Soit $\alpha = (\alpha_i)_{i \in \{1,\cdots,N\}}$ les multiplicateurs de Lagrange (variables duales), avec $N = |\Pi_a|$

$$L(w, w_0, \alpha) = \frac{1}{2} w' \cdot w - \sum_{i=1}^{N} \alpha_i \left[C(x_i) \left(w' \cdot x_i + w_0 \right) - 1 \right]$$

- **Optimisation Lagrangienne**

Le problème primal et sa formulation duale ont la même solution qui correspond a un point-selle du Lagrangien.
w^* et w_0^* vérifie donc :

$$\frac{\partial L}{\partial w}(w^*, w_0^*, \alpha^*) = \frac{\partial L}{\partial w_0}(w^*, w_0^*, \alpha^*) = \frac{\partial L}{\partial \alpha}(w^*, w_0^*, \alpha^*) = 0$$

Conditions suffisantes pour l'optimum si le Lagrangien est convexe.

$\frac{\partial L}{\partial w}(.) = 0 \Rightarrow w^* = \sum_{i=1}^{N} \alpha_i^* C(x_i) x_i$ $\qquad$ $\frac{\partial L}{\partial w_0}(.) = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i^* C(x_i) = 0$

Optimisation de la marge (3) : programme dual

Maximisation du Lagrangien (simplifié)

$$\left| \begin{array}{l} \max_\alpha L(w^*, w_0^*, \alpha) = \sum_{i=1}^{N} \alpha_i^2 - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j C(x_i) C(x_j) \left(x_i' \cdot x_j \right) \\ \quad \alpha \geq 0 \\ \quad \sum_{i=1}^{N} \alpha_i C(x_i) = 0 \end{array} \right.$$

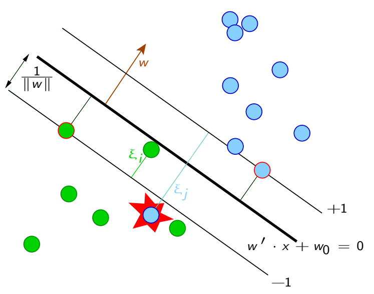
- Seuls les vecteurs supports auront des $\alpha_i^* > 0$! Problème quadratique de petite taille
- Le programme dual ne s'exprime que sur les données (N intervient mais pas d) !
- $w^* = \sum_{i=1}^{N} \alpha_i^* C(x_i) x_i \Rightarrow w^*$ uniquement en fonction des vecteurs supports !
- w_0^* ? $\forall i \in \{1, \cdots, N\}, \alpha_i^* \left(C(x_i)(w^{*\prime} \cdot x_i + w_0^*) - 1 \right) = 0$

Équation de l'hyperplan séparateur en fonction de Π_a

$$0 = H(x) = w^{*\prime} \cdot x + w_0^* = \sum_{x_i \in \Pi_a} \alpha_i^* C(x_i) \left(x_i' \cdot x \right) + w_0^*$$

- Classification : $\forall x, \widehat{C}(x) = \sigma(H(x))$ (seuls les vecteurs supports sont nécessaires !)

Données *presque* linéairement séparables ?



- Éléments mal classés :
  $$(C(x_j).(w' \cdot x_j + w_0) < 0)$$
- Éléments ne respectant pas la marge :
  $$|w' \cdot x_j + w_0| < 1.$$

$\Rightarrow$ introduction de *slack variables* $\xi_i \geq 0$:

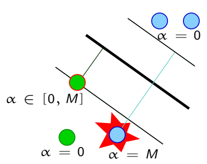
Modification du primal

$$\left| \begin{array}{l} \min_w \frac{1}{2} \|w\|^2 + M \sum_{i=1}^{N} \xi_i \quad (M \geq 0) \\ \quad \forall x \in \Pi_a, C(x) \left(w' \cdot x + w_0 \right) \geq 1 - \xi_i \end{array} \right.$$

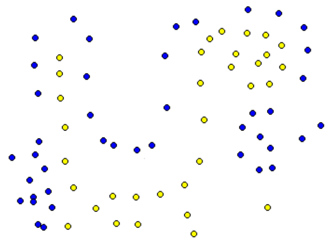
M = *trade-off* entre minimisation de la marge et minimisation des erreurs de classification.

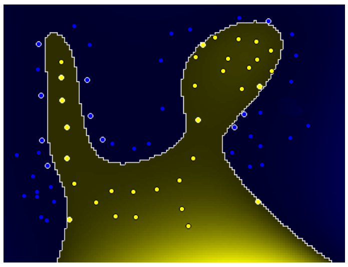
Modification du dual

$$\left| \begin{array}{l} \max_\alpha \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j C(x_i) C(x_j) \left(x_i' \cdot x_j \right) \\ \quad \forall i, 0 \leq \alpha_i \leq M \\ \quad \sum_{i=1}^{N} \alpha_i C(x_i) = 0 \end{array} \right.$$



Données pas du tout linéairement séparable ?





Idée : l'expression obtenue précédemment pour les SVM montre une méthode peu sensible à la dimension d de l'espace.

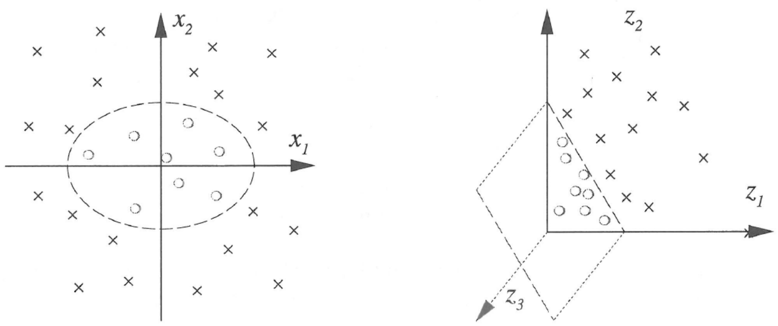
Redescription Φ

Modifier l'espace de description $\mathcal{D}$ en un espace de plus haute dimension (éventuellement infinie) doit permettre de rendre plus probable la séparation linéaire.

$$\Phi : \left| \begin{array}{ccc} \mathcal{D} & \longrightarrow & \Phi(\mathcal{D}) \subseteq \mathbb{R}^D \\ x & \mapsto & \Phi(x) = \begin{bmatrix} \Phi_1(x) \\ \cdots \\ \Phi_D(x) \end{bmatrix} \end{array} \right.$$

Φ est non linéaire et D est très grand (voir infini).
$\Phi(\mathcal{D})$ est l'espace de redescription.

Redescription Φ : un exemple



From : *Learning with Kernels* – B.Schölkopf and A.J.Smola – p29

- Dans l'espace (x_1, x_2), les données ne sont pas linéairement séparables.
- Dans l'espace (z_1, z_2, z_3) avec la transformation $\Phi(x_1, x_2) = (x_1^2, x_2^2, \sqrt{2} \cdot x_1 \cdot x_2)$, l'ellipse se transforme en hyperplan (parallèle à l'axe z_3), $\Phi(\Pi_a)$ est linéairement séparable.

Optimisation dans l'espace $\Phi(\mathcal{D})$

Remarque : On peut utiliser cette redescription avec toute forme de classifieur. Par exemple :

- Analyse discriminante linéaire
- Perceptrons
- Machine à vecteurs de support
- Analyse en composantes principales
- etc.

Nous nous intéressons ici au SVM.

Redescription de SVM

- $$\left| \begin{array}{l} \max_\alpha L(w^*, w_0^*, \alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j C(x_i) C(x_j) \, (\Phi(x_i)' \cdot \Phi(x_j)) \\ \quad \alpha \geq 0 \\ \quad \sum_{i=1}^N \alpha_i C(x_i) = 0 \end{array} \right.$$
- $0 = H(x) = {w^*}' \cdot x + w_0^* = \sum_{x_i \in \Pi_a} \alpha_i^* C(x_i) \, (\Phi(x_i)' \cdot \Phi(x)) + w_0^*$

$(\Phi(.)' \cdot \Phi(.))$ risque d'être très long à calculer !
(si $D >>$ ou si Φ est une fonction compliquée ...)

Fonction noyau et *Kernel trick*

Définition (Fonction Noyau)

$$K : \left| \begin{array}{ccc} \mathcal{D} \times \mathcal{D} & \longrightarrow & \mathbb{R} \\ (x, y) & \mapsto & (\Phi(x)' \cdot \Phi(y)) \end{array} \right. \text{ est le noyau (kernel) de } \Phi$$

SVM avec kernel

On peut alors calculer (α^*, w^*, w_0^*) sans connaître Φ ni même sa dimension :

- $$\left| \begin{array}{l} \max_\alpha L(w^*, w_0^*, \alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j C(x_i) C(x_j) K(x_i, x_j) \\ \quad \alpha \geq 0 \\ \quad \sum_{i=1}^N \alpha_i C(x_i) = 0 \end{array} \right.$$
- $0 = H(x) = {w^*}' \cdot x + w_0^* = \sum_{x_i \in \Pi_a} \alpha_i^* C(x_i) K(x_i, x) + w_0^*$

Propriétés de $K(.,.)$

- K est continue
- K est symétrique : $K(x, y) = K(y, x)$
- K est semi-définie positive : $\forall (x_i)_\mathcal{I} \in \mathcal{D}^\mathcal{I}, \forall (c_i)_\mathcal{I} \in \mathbb{R}^\mathcal{I}, \sum_{(i,j) \in \mathcal{I}^2} c_i c_j K(x_i, x_j) \geq 0$

La fonction noyau K est une mesure de similarité.

Fonction noyau et *Kernel trick* (2)

K facilite le calcul de l'optimum du problème dual mais on doit toujours passer par Φ pour calculer K.

Théorème (Mercer)

Si $K : \mathcal{D} \times \mathcal{D} \longrightarrow \mathbb{R}$ est symétrique et semi-définie positive alors $\exists \Phi : \mathcal{D} \longrightarrow \mathbb{R}^D$ telle que $K(x, x') = \Phi(x)' \cdot \Phi(x')$.

On peut alors utiliser K comme fonction noyau.

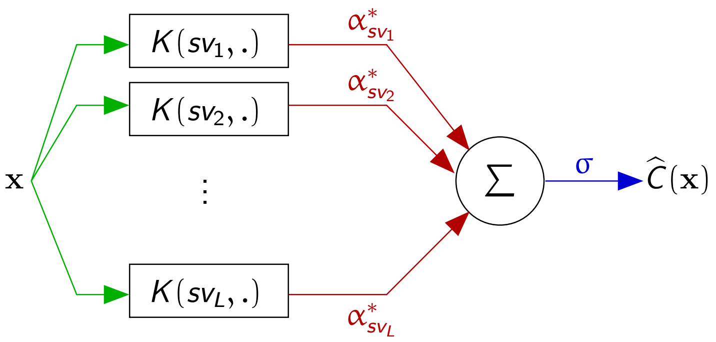
⚠ Ce n'est pas un théorème constructif : on ne connait pas Φ.

- On peut donc se passer de Φ si on connait K vérifiant les bonnes hypothèses.
- SVM avec noyau travaille implicitement dans l'espace de redescription.
- **Démarche** :
  - choisir une mesure de similarité sémantiquement dépendante du domaine,
  - vérifier qu'elle possède les propriétés nécessaires (symétrie, semi-définie positive).
  - construire et résoudre le problème duale en utilisant cette mesure si OK.

Classification à l'aide de noyau

Les seuls α_i^* non nuls sont ceux des vecteurs supports (exemples critiques) :

$$\widehat{C}(x) = \sigma \left(\sum_{x_i \in \text{SupportVectors}(\Pi_a, \mathbf{w}^*)} \alpha_i^* C(x_i) K(x_i, x) + w_0^* \right)$$



Générations de fonctions noyaux

Propriétés

Avec $\mathcal{D} \subset \mathbb{R}^d$,

1. $\forall K_1, K_2 : \mathcal{D}^2 \longrightarrow \mathbb{R}$ deux noyaux,
   1. $K = K_1 + K_2$ est un noyau
   2. $\forall a \in \mathbb{R}^+, K = aK_1$ est un noyau
   3. $K = K_1 \cdot K_2$ est un noyau
   4. $\forall p(x)$ polynôme à coefficients positifs, $K(x,y) = p(K_1(x,y))$ est un noyau
   5. $K(x,y) = e^{K_1(x,y)}$ est un noyau
2. $\forall f : \mathcal{D} \longrightarrow \mathbb{R}, K(x,y) = f(x) \cdot f(y)$ est un noyau
3. $\forall K_3 : \mathbb{R}^m \times \mathbb{R}^m \longrightarrow \mathbb{R}$ noyau et $\varphi : \mathcal{D} \longrightarrow \mathbb{R}^m$,
   $K(x,y) = K_3(\varphi(x), \varphi(y))$ est un noyau
4. $\forall B$ matrice $d \times d$ symétrique, semi-définie positive,
   $K(x,y) = x' \cdot B \cdot y$ est un noyau

Exemple

Proposition

$$K(x,y) = e^{-\frac{\|x-y\|^2}{\sigma^2}} \text{ est un noyau.}$$

Démonstration :

- $\|x-y\|^2 = \|x\|^2 + \|y\|^2 - 2(x' \cdot y)$
- $\Rightarrow K(x,y) = e^{\frac{-\|x\|^2}{\sigma^2}} \cdot e^{\frac{-\|y\|^2}{\sigma^2}} \cdot e^{\frac{2(x' \cdot y)}{\sigma^2}}$
- mais alors $K(x,y) = K_a(x,y) \cdot K_b(x,y)$ avec
  - $K_a(x,y) = e^{\frac{-\|x\|^2}{\sigma^2}} \cdot e^{\frac{-\|y\|^2}{\sigma^2}}$: noyau par la propriété 2 !
  - $K_b(x,y) = e^{\frac{2(x' \cdot y)}{\sigma^2}}$
    - I_n symétrique, semi-définie positive $\Rightarrow x' \cdot I_n \cdot y$ est un noyau (propriété 4)
    - $\Rightarrow \frac{2(x' \cdot y)}{\sigma^2}$ est un noyau (propriété 1.2)
    - $\Rightarrow K_b(x,y)$ est un noyau (propriété 1.5)
- $\Rightarrow K(x,y) = K_a(x,y) \cdot K_b(x,y)$ est un noyau (propriété 1.3)

■

Reconstruction de Φ

Par le théorème de Mercer, si $K(.,.)$ vérifie les bonnes propriétés, Φ n'a jamais à être explicitée... Mais pour le fun (et la compréhension) ...

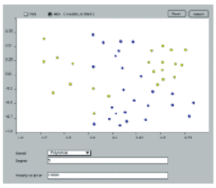
Rappel :Φ est telle que $K(x,y) = (\Phi(x)' \cdot \Phi(y))$

Exemple de reconstruction de Φ

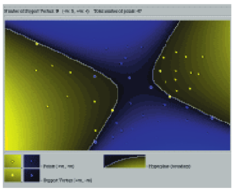
$K_n(x,y) = (x' \cdot y)^n$ est un noyau.

Trouver Φ_n ?

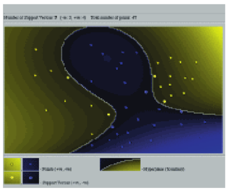
Influence du noyau



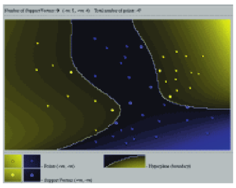
- **47 exemples (22 +, 25 -)**
- ***Exemples critiques* : 4 + et 3 -**



(5-, 4+)

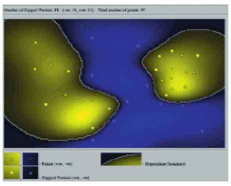


(3-, 4+)

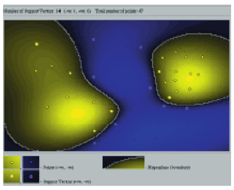


(5-, 4+)

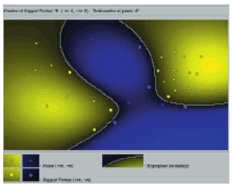
Ici *fonction polynomiale* de *degré 2, 5, 8* et *C = 10000*



(10-, 11+)



(8-, 6+)



(4-, 5+)

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

from : Laurent Miclet