MA431 : Mathématiques appliquées à la sécurité

Classification : Support Vector Machines(SVM)

D. Barcelo

Grenoble INP ESISAR

2022/2023

SVM

SVM

Les techniques SVM (*Support Vector Machines* ou *machines à vecteurs de support* ou *séparateur à vaste marge*) sont des techniques

- **de classification** :
- **supervisées** :

SVM

SVM

Les techniques SVM (*Support Vector Machines* ou *machines à vecteurs de support* ou *séparateur à vaste marge*) sont des techniques

- **de classification** :
  Variable Y à étudier/prévoir discrète, voire binaire
- **supervisées** :

SVM

SVM

Les techniques SVM (*Support Vector Machines* ou *machines à vecteurs de support* ou *séparateur à vaste marge*) sont des techniques

- **de classification** :
  Variable Y à étudier/prévoir discrète, voire binaire
- **supervisées** :
  besoin d'une base d'apprentissage et d'une base test

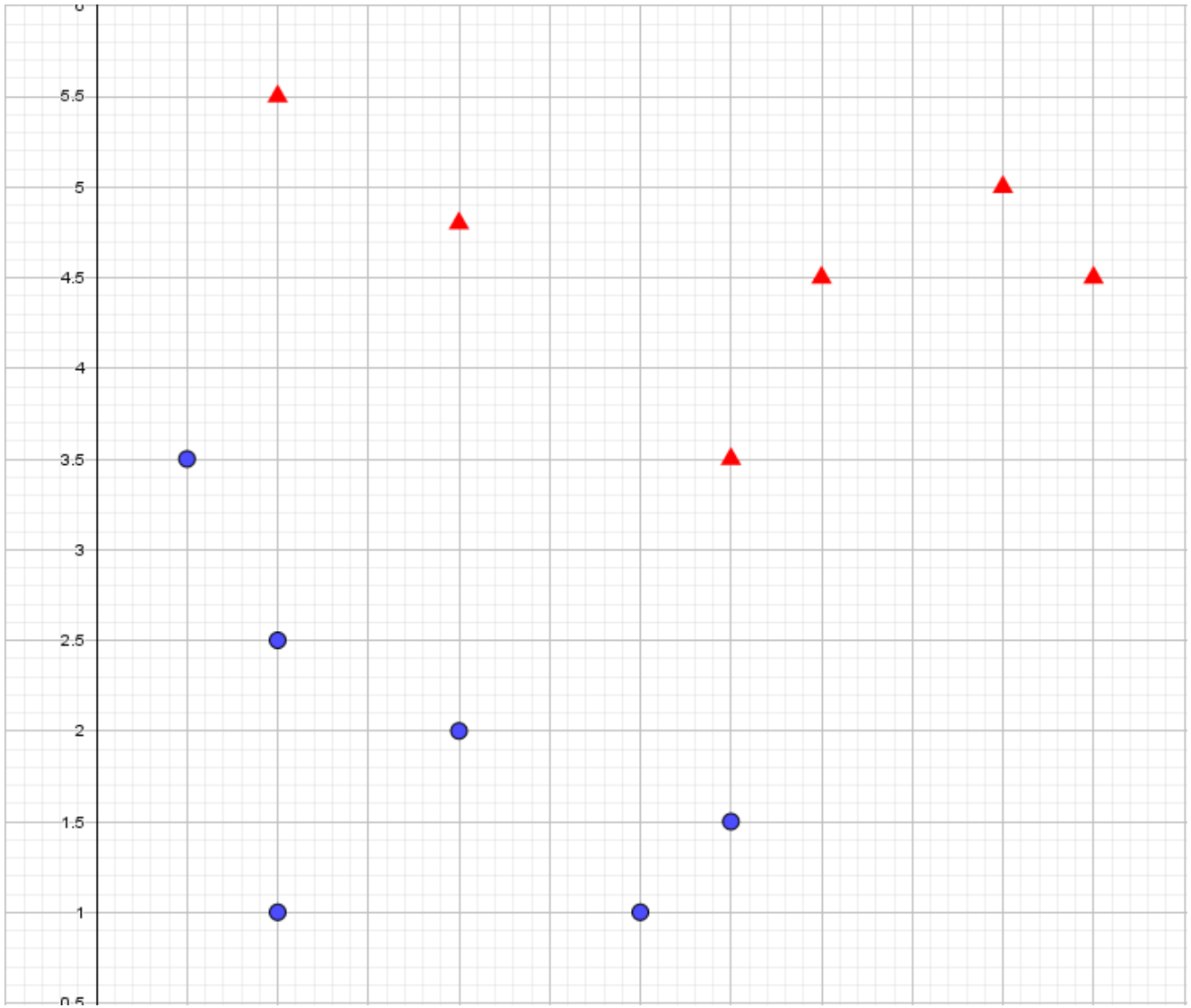
Modélisation

On observe des données assimilées à des vecteurs x_i de dimension n.
Les données sont étiquetées en deux groupes : $Y(\Omega) = \{-1; +1\}$.
Objectif :

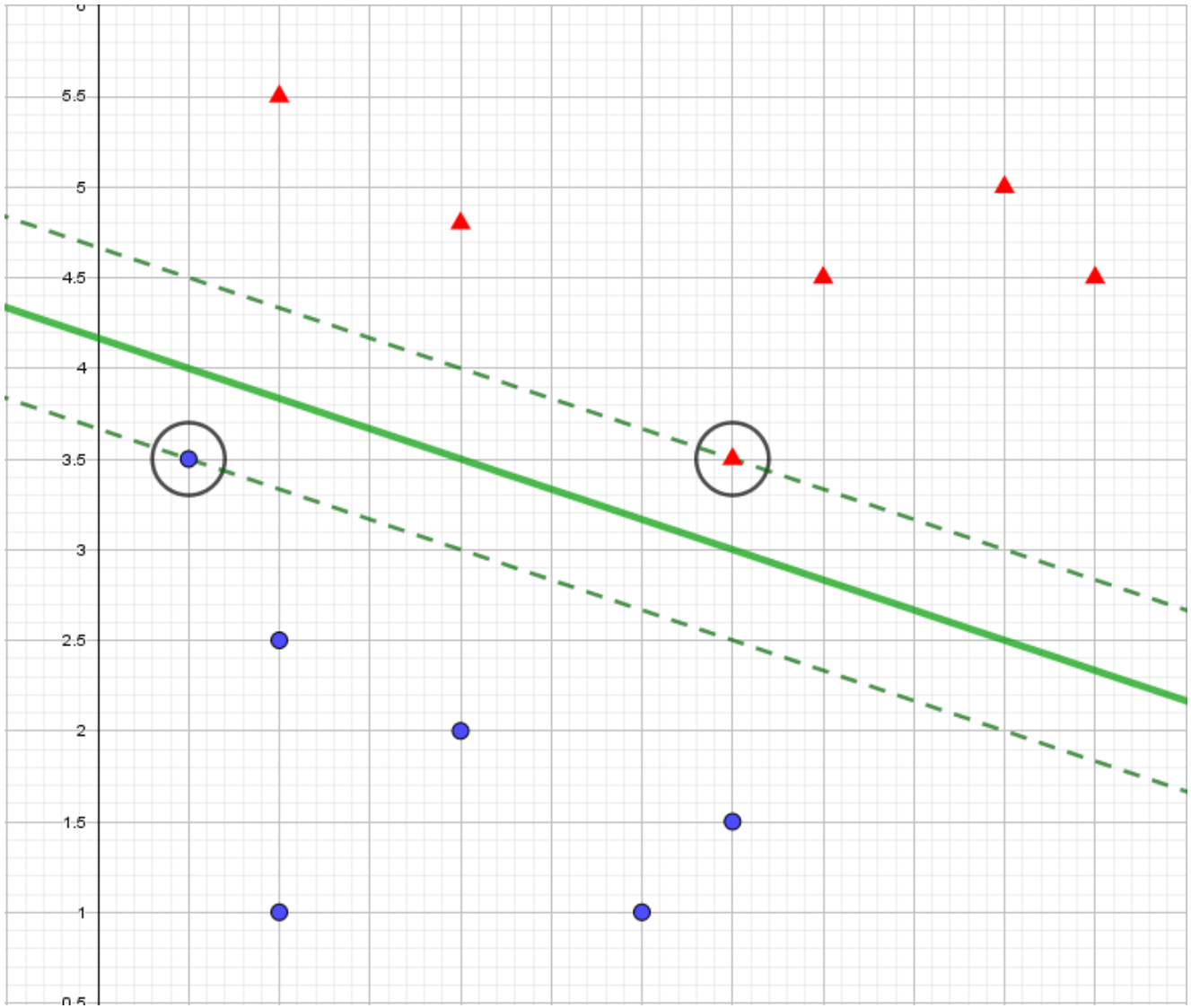
- Classer les points $\{(x_i, y_i) \in \mathbb{R}^d \times \{-1, 1\}\}_{i=1}^n$.
- **Séparation par un hyperplan.**
  Construire un classifieur linéaire : $f(x) = {}^t x \bullet a + b$ avec $a \in \mathbb{R}^d$ et $b \in \mathbb{R}$.
  a représente un vecteur normal à l'hyperplan.
- Règle de classification :

$$\hat{y}(x) = \begin{cases} 1 \text{ si } f(x) > 0 \\ -1 \text{ si } f(x) < 0 \end{cases}.$$

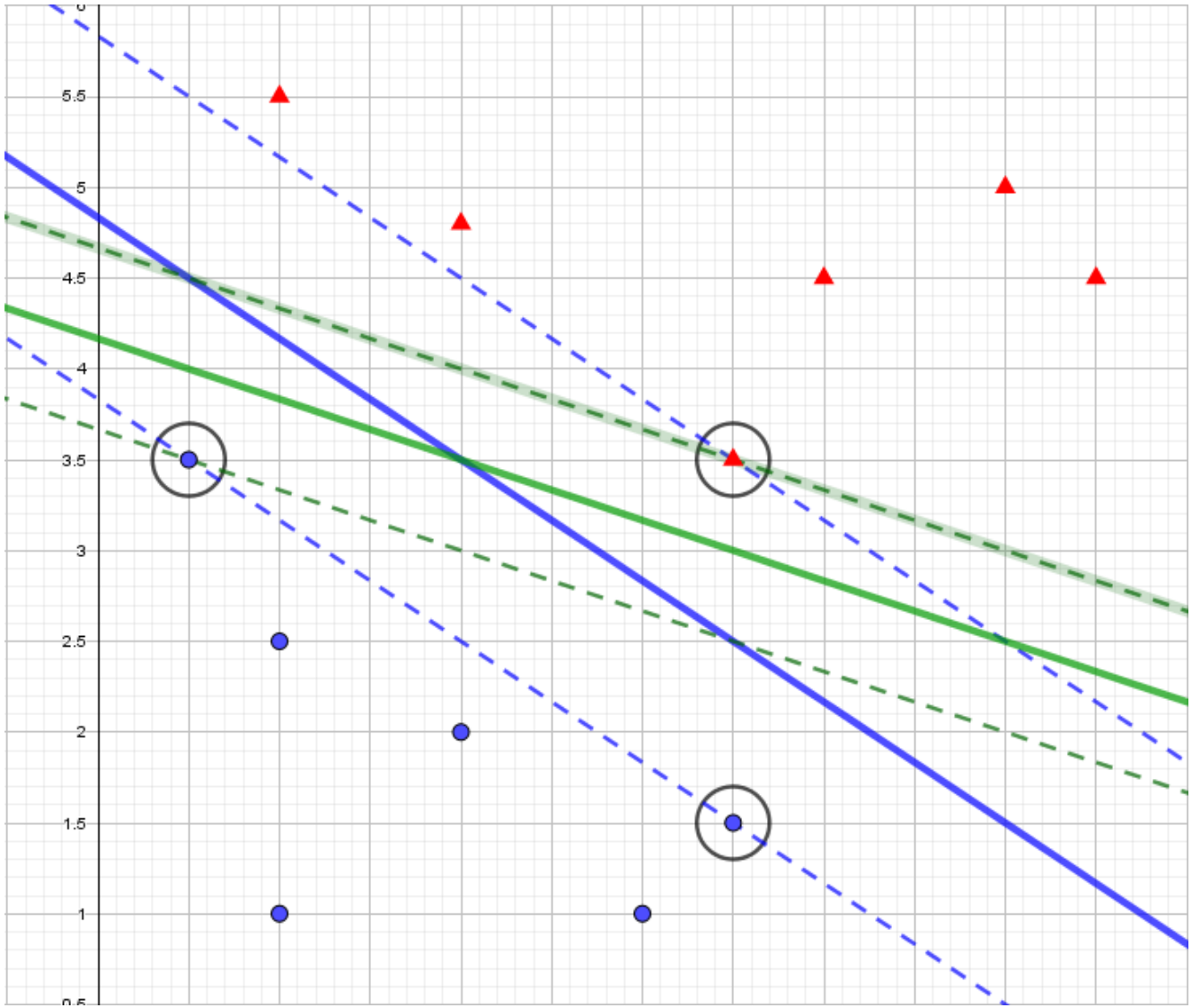
Principes



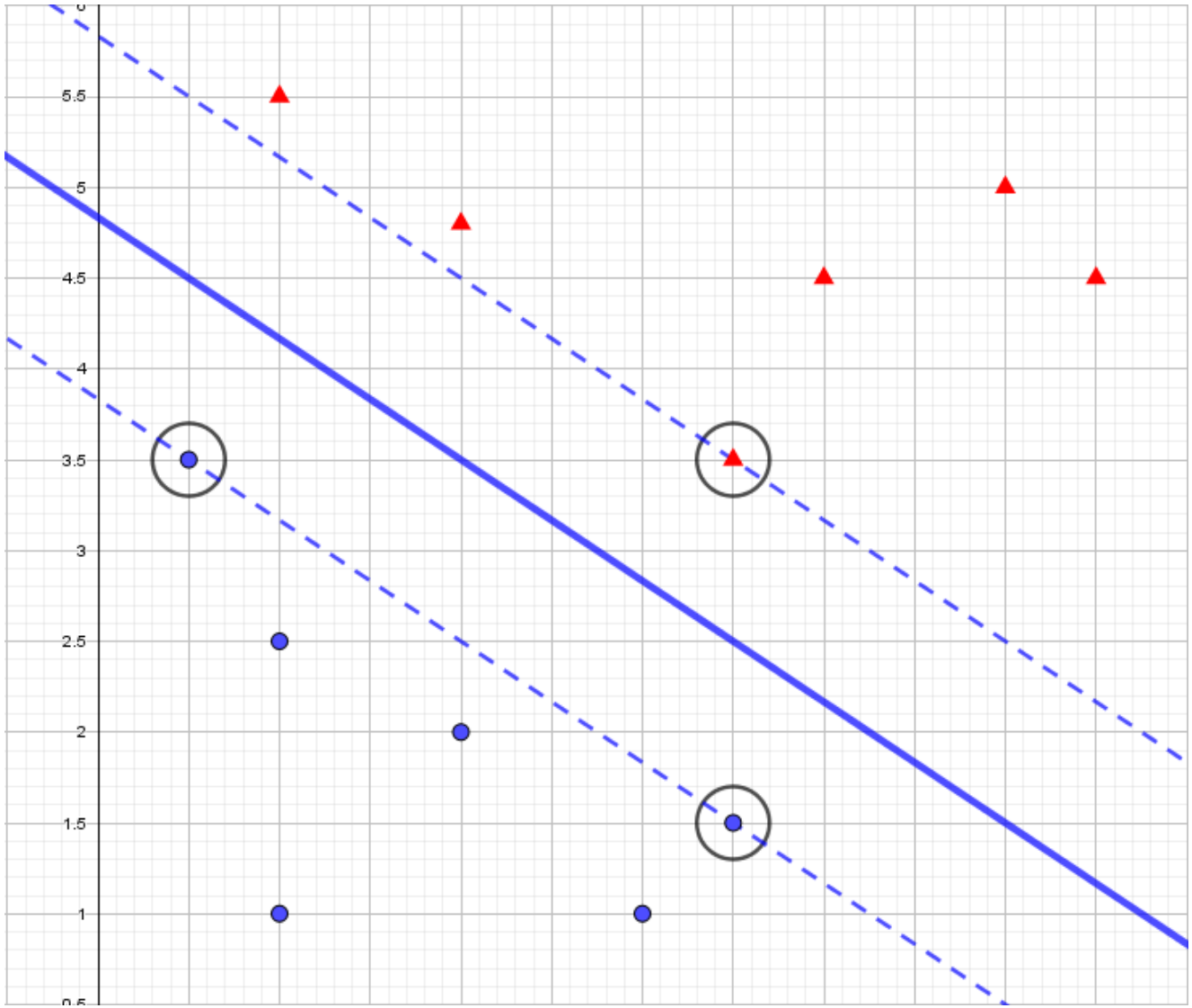
Principes



Principes



Principes



Principes

Principes du séparateur :

- **Bon ajustement** du modèle :
- **Robustesse** du modèle :

Principes

Principes du séparateur :

- **Bon ajustement** du modèle :
  l'hyperplan sépare bien les groupes à discriminer.
- **Robustesse** du modèle :

.

Principes

Principes du séparateur :

- **Bon ajustement** du modèle :
  l'hyperplan sépare bien les groupes à discriminer.
- **Robustesse** du modèle :
  l'hyperplan est le plus loin possible de toutes les observations.
- Hyperplan optimal qui maximise la marge.

Principes

Géométriquement :

- **Equation de l'hyperplan H :** ${}^t x \bullet a + b = 0$
- **Distance de x à H :** $d(x, H) = \dfrac{|{}^t x \bullet a + b|}{\|a\|}$.
- **Hyperplans de la marge :** ${}^t x \bullet a + b = 1$ et ${}^t x \bullet a + b = -1$.
- **Largeur de la marge :** $\dfrac{2}{\|a\|}$.

  Résolution d'un problème d'optimisation sous contrainte.

Principes

Principes du séparateur mathématiquement :

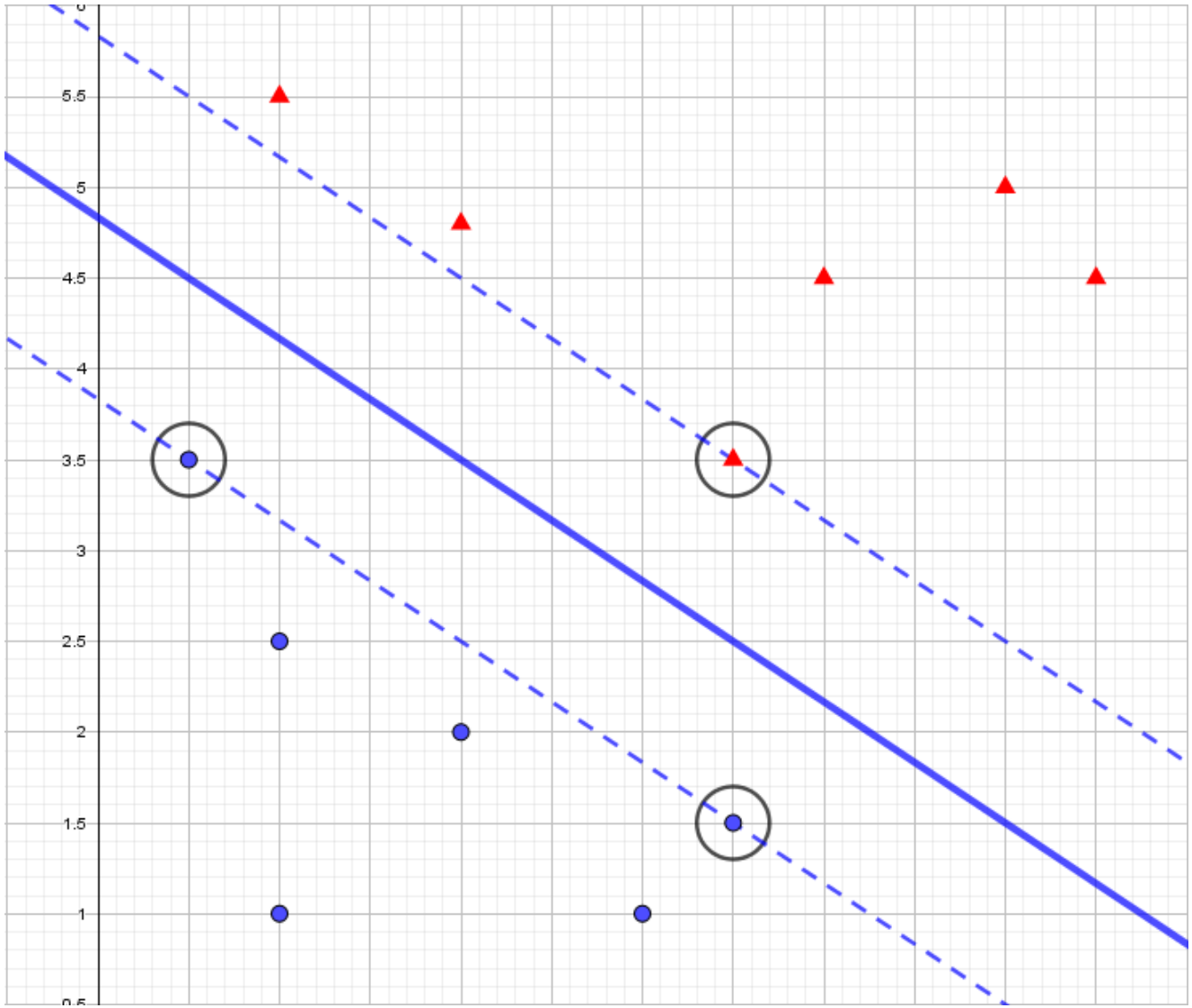
- **Bon ajustement du modèle : l'hyperplan sépare bien les groupes à discriminer.** $\forall i,\ y_i f(x_i) = 1$
- **l'hyperplan est le plus loin possible de toutes les observations :** $\max \dfrac{|{}^t x \bullet a + b|}{\|a\|}$.
- **Hyperplan optimal qui maximise la marge :** $\max \dfrac{2}{\|a\|}$.

  Résolution d'un problème d'optimisation sous contrainte.

Exemple en dimension 2

- On observe dispose de n observations $(X_i)_{i\in[\![1;n]\!]}$ de deux variables quantitatives
- X_i est un point de coordonnées (x_i, y_i)
- Z est la variable qui représente la classe de X_i. $Z\Omega) = \{-1, +1\}$

Exemple en dimension 2



Objectif

On veut déterminer l'hyperplan H défini par l'équation : $a_1x + a_2y + b = 0$.

- L'hyperplan de séparation inférieur a pour équation : $a_1x + a_2y + b = -1$.
- L'hyperplan de séparation supérieur a pour équation : $a_1x + a_2y + b = 1$.
- La marge a pour valeur $m = \dfrac{2}{\sqrt{a_1^2 + a_2^2}} = \dfrac{2}{\|a\|}$ où $a = (a_1, a_2)$.

Optimisation

On est donc ramené à un problème d'optimisation :

- Maximiser $m = \dfrac{2}{\|a\|}$.
- Sous la contrainte :
  $\forall i \in [\![1; n]\!]\ z_i\,(a_1 x_i + a_2 y_i + b) \geqslant 1.$

Minimisation

On est donc ramené à un problème d'optimisation :

- Minimiser $\dfrac{2}{m^2} = \dfrac{\|a\|^2}{2}$.
- Sous la contrainte :
  $\forall i \in [\![1; n]\!]\ 0 \geqslant 1 - z_i\left(a_1 x_i + a_2 y_i + b\right)$.

Multiplicateurs de Lagrange

Optimisation à l'aide des multiplicateurs de Lagrange : On pose :

$$L(a_1, a_2, b, \lambda) = \frac{a_1^2 + a_2^2}{2} + \sum_{i=1}^{n} \lambda_i (1 - z_i (a_1 x_i + a_2 y_i + b))$$

On détermine les extrema de L.

◂ Rappels ◂ Calculs

Optimisation

Maximiser : $L(\lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j z_i z_j \, (X_i \bullet X_j)$

Sous la contrainte : $\sum_{i=1}^{n} \lambda_i z_i = 0.$

Fonction score solution

Après optimisation et détermination des λ_i, on pose :

$f(X) = a_1 x_1 + a_2 x_2 + b$

ou :

$$f(X) = \sum_{i=1}^{n} \lambda_i z_i \, (X_i \bullet X)$$

Vecteurs supports

De la définition de L et de la condition suivante :

$$L(a_1, a_2, b, \lambda) = \frac{a_1^2 + a_2^2}{2} + \sum_{i=1}^{n} \lambda_i (1 - z_i (a_1 x_i + a_2 y_i + b))$$

$$\forall i \in [\![1; n]\!] \; 1 - z_i (a_1 x_i + a_2 y_i + b) = 0$$

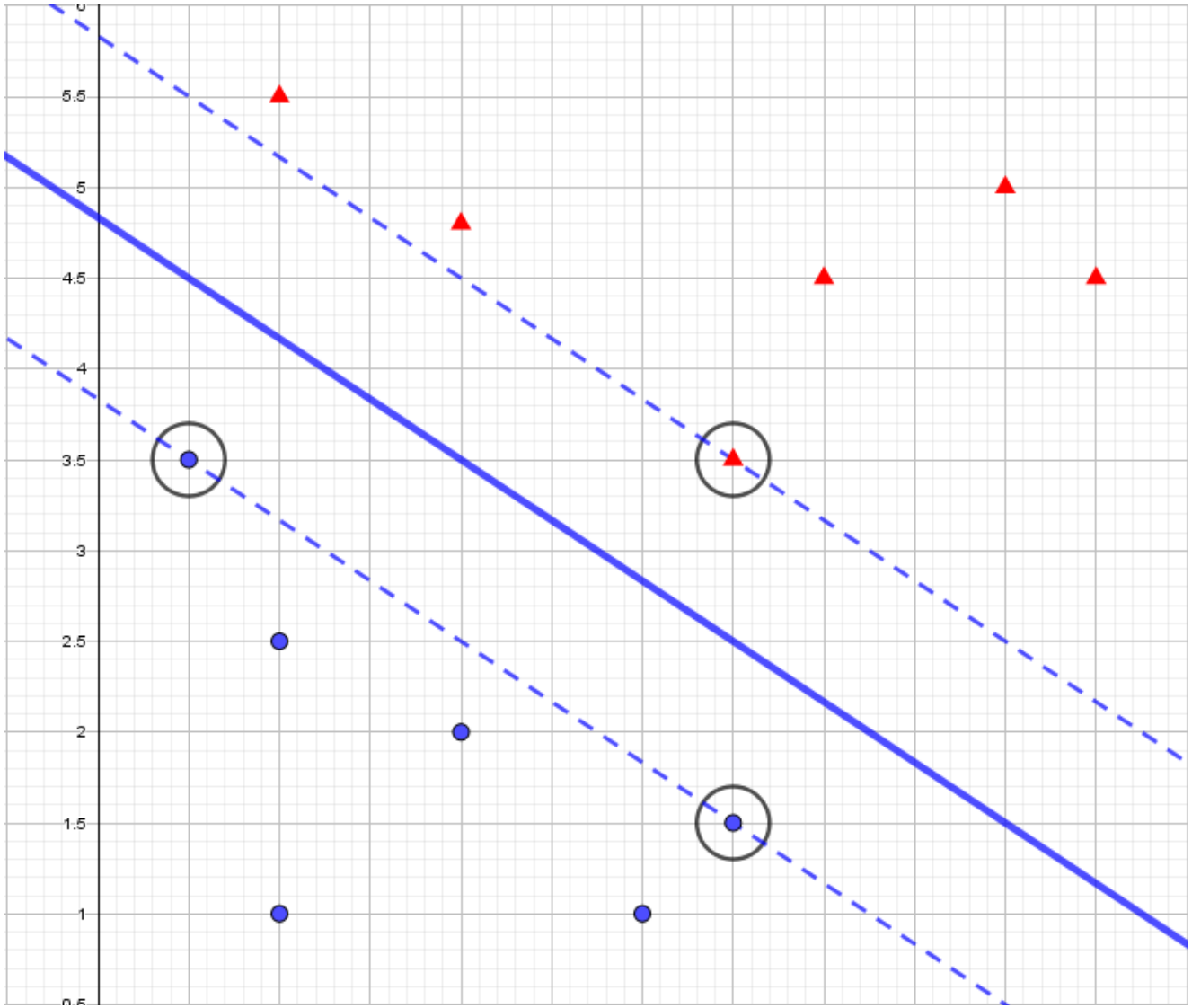
On peut déduire que si $\lambda_i \neq 0$ alors $1 - z_i (a_1 x_i + a_2 y_i + b) = 0$.

Vecteurs supports

Les $(\lambda)_i$ sont donc :

- soit nuls et ils correspondent aux X_i qui sont bien classés (à l'extérieur de la marge)
- soit non nuls et ils correspondent aux X_i qui se trouvent sur la marge et permettent de la définir.
  Il s'agit des vecteurs supports.

Vecteurs supports



Vecteurs supports

Il y a exactement s vecteurs supports :

$$f(X) = \sum_{i=1}^{s} \lambda_i z_i \left(X_i \bullet X\right)$$

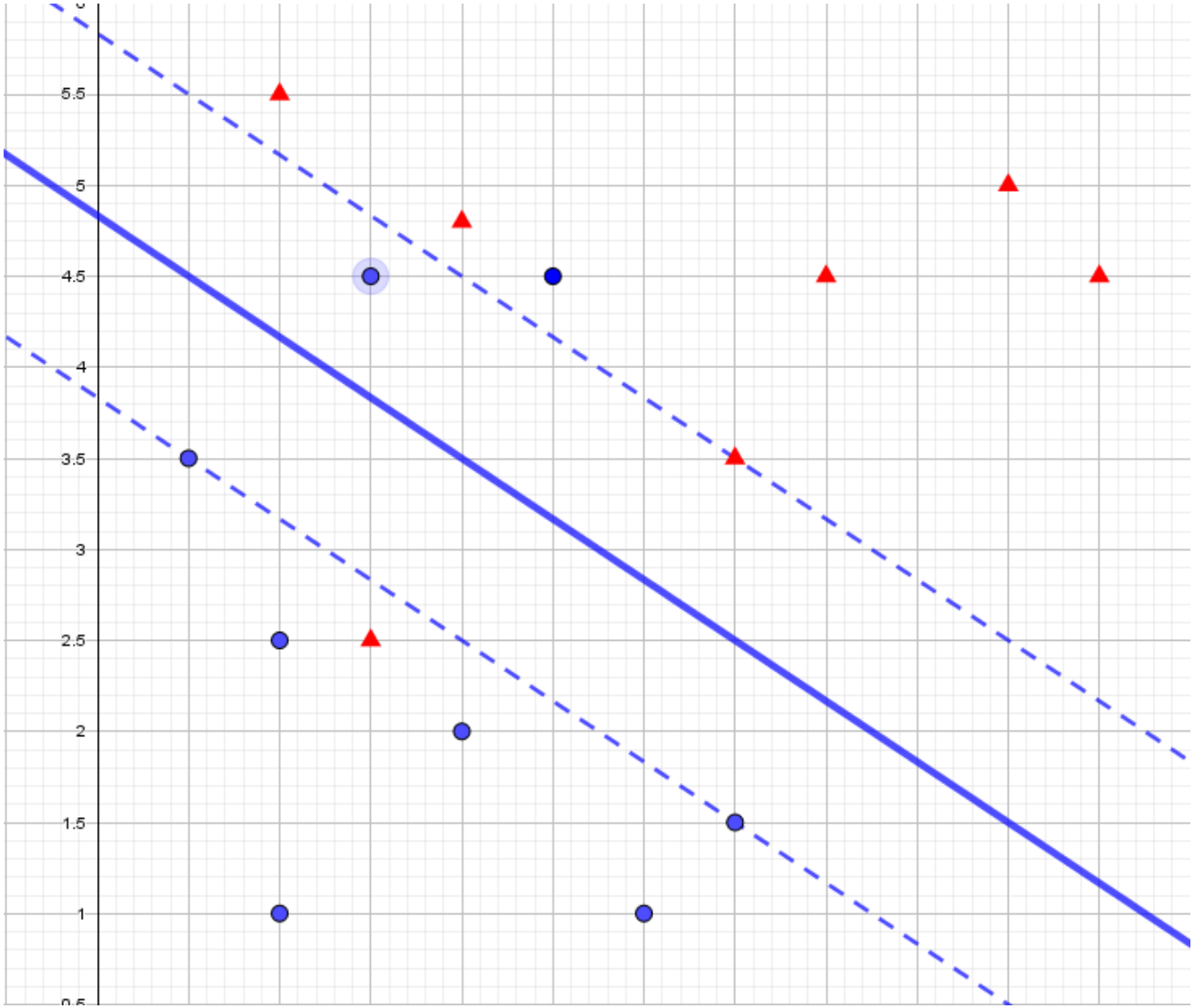
Le modèle est donc entièrement déterminé par quelques vecteurs.

Relaxation

En pratique, on doit autoriser des erreurs de classement pour améliorer la marge.

- On introduit des variables de relaxation (slack variables)
- Soit $i \in [\![1, n]\!]$ ε_i représente l'erreur de classement de X_i.
- Si $\varepsilon_i = 0$ pas d'erreur.
- Si $\varepsilon_i > 1$ Erreur et position au-delà de la marge.
- Si $1 \geqslant \varepsilon_i > 0$ Erreur et position à l'intérieur de la marge.
- La contrainte devient $\forall i \in [\![1; n]\!]$ $z_i (a_1 x_i + a_2 y_i + b) \geqslant 1 - \varepsilon_i$.
- Il faut trouver un compromis entre maximisation de la marge et contrôle des erreurs.

Relaxation



Relaxation

Le problème d'optimisation devient :

- Maximiser $m = \dfrac{2}{\|a\|} + C \sum_{i=1}^{n} \varepsilon_i$.
- Sous la contrainte :
  $\forall i \in [\![1; n]\!]\ z_i \left(a_1 x_i + a_2 y_i + b\right) \geqslant 1 - \varepsilon_i$.

où C est un paramètre de coût à fixer.

Relaxation

Version duale :

Maximiser : $L(\lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \lambda_i \lambda_j z_i z_j \, (X_i \bullet X_j)$

Sous les contraintes :

$$\forall i \in [\![1; n]\!] \; C \geqslant \lambda_i \geqslant 0$$
$$\sum_{i=1}^{n} \lambda_i z_i = 0.$$

Relaxation

- C est un paramètre de coût des erreurs.
- Plus C est grand, plus grande est la sensibilité aux erreurs.
- C peut être déterminé par validation croisée.
- Attention à équilibrer ajustement et robustesse.

Bilan

Pour construire un modèle de SVM, il faut :

- disposer d'une base d'apprentissage (et d'une base test).
- Choisir le paramètre C
- Résoudre le problème dual pour obtenir les λ_i.
- En déduire a et b.

Exemple 1

Exemple 1

Astuces du SVM

Les astuces :

- Transformation d'un problème non séparable linérairement en un problème séparable linéairement.

- Calculs en dimension supérieure.

Astuces du SVM

Les astuces :

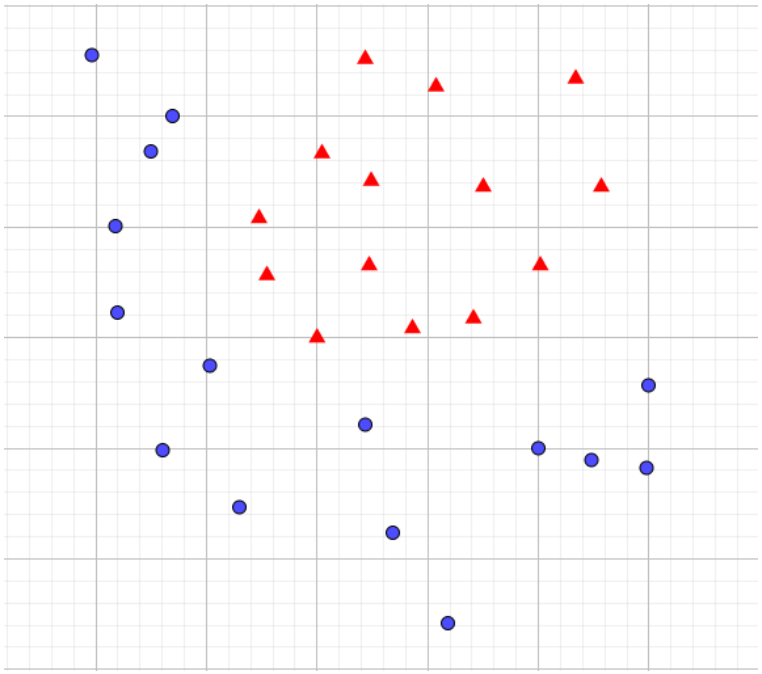
- Transformation d'un problème non séparable linérairement en un problème séparable linéairement.
  l'astuce du noyau ou kernel trick
- Calculs en dimension supérieure.

Astuces du SVM

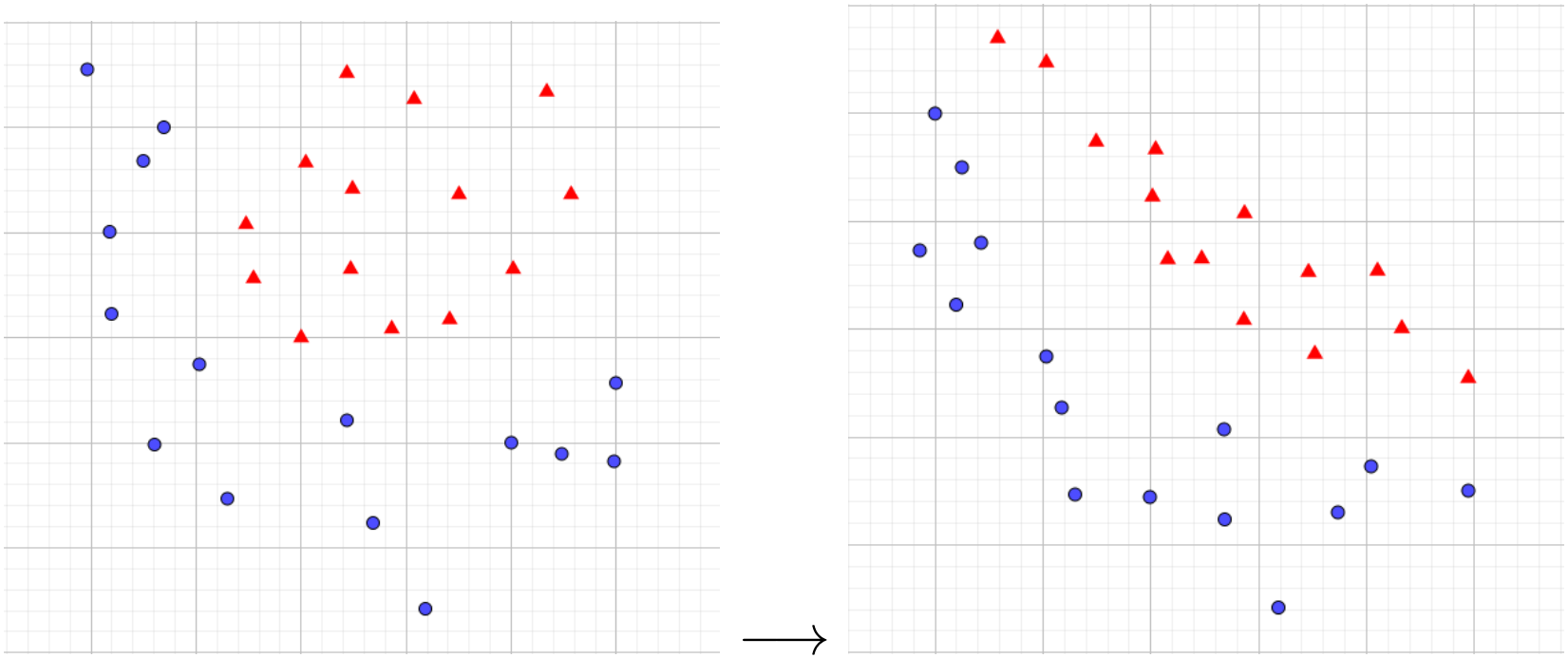
Les astuces :

- Transformation d'un problème non séparable linérairement en un problème séparable linéairement.
  l'astuce du noyau ou kernel trick
- Calculs en dimension supérieure.
  la malédiction de la dimension devient la bénédiction de la dimension !

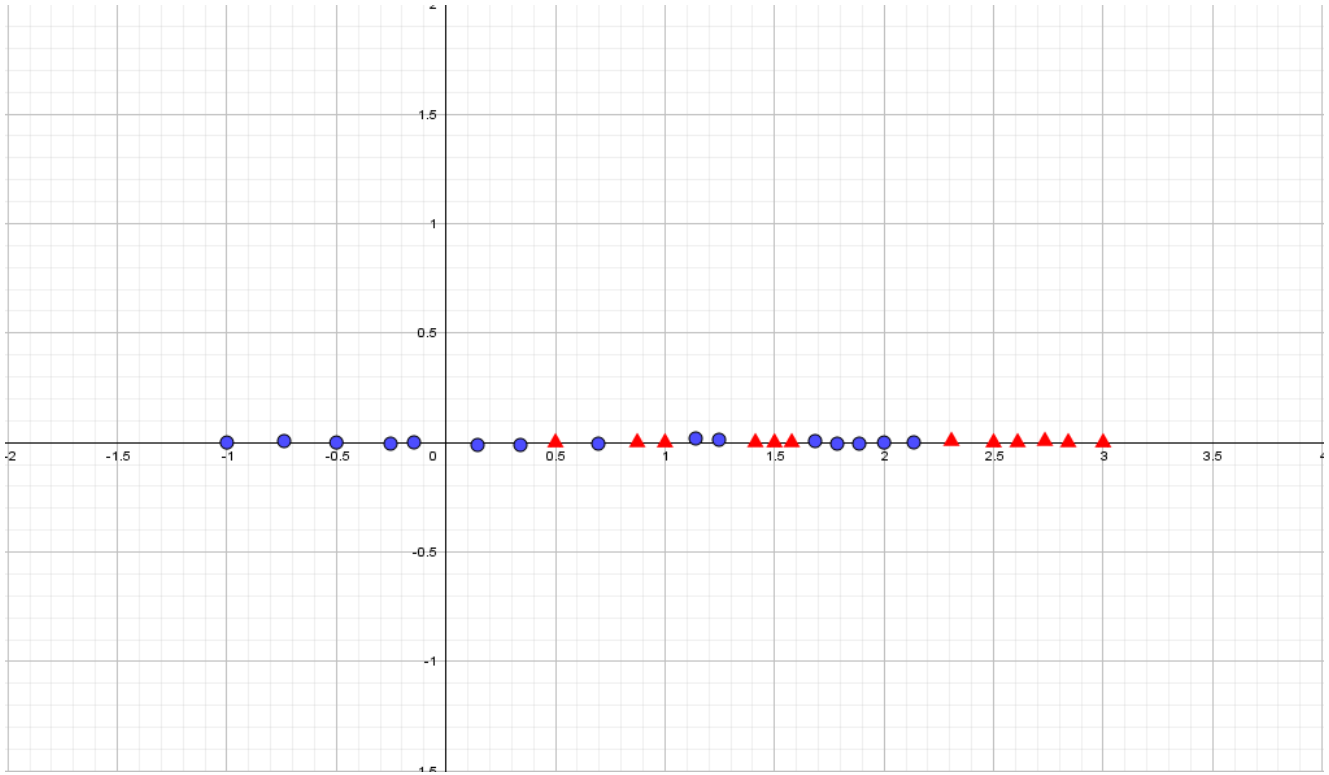
Kernel trick



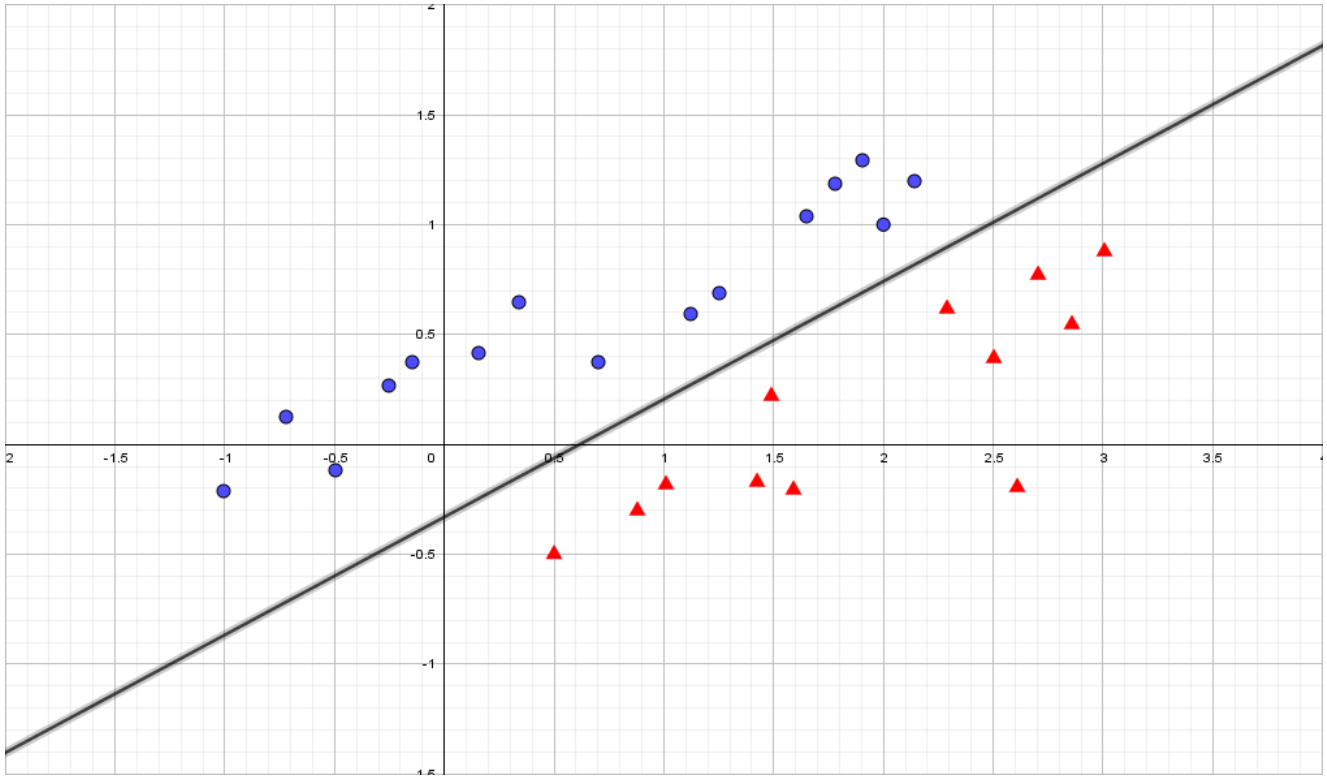
Kernel trick



Dimension supérieure



Dimension supérieure



Dimension supérieure et kernel trick

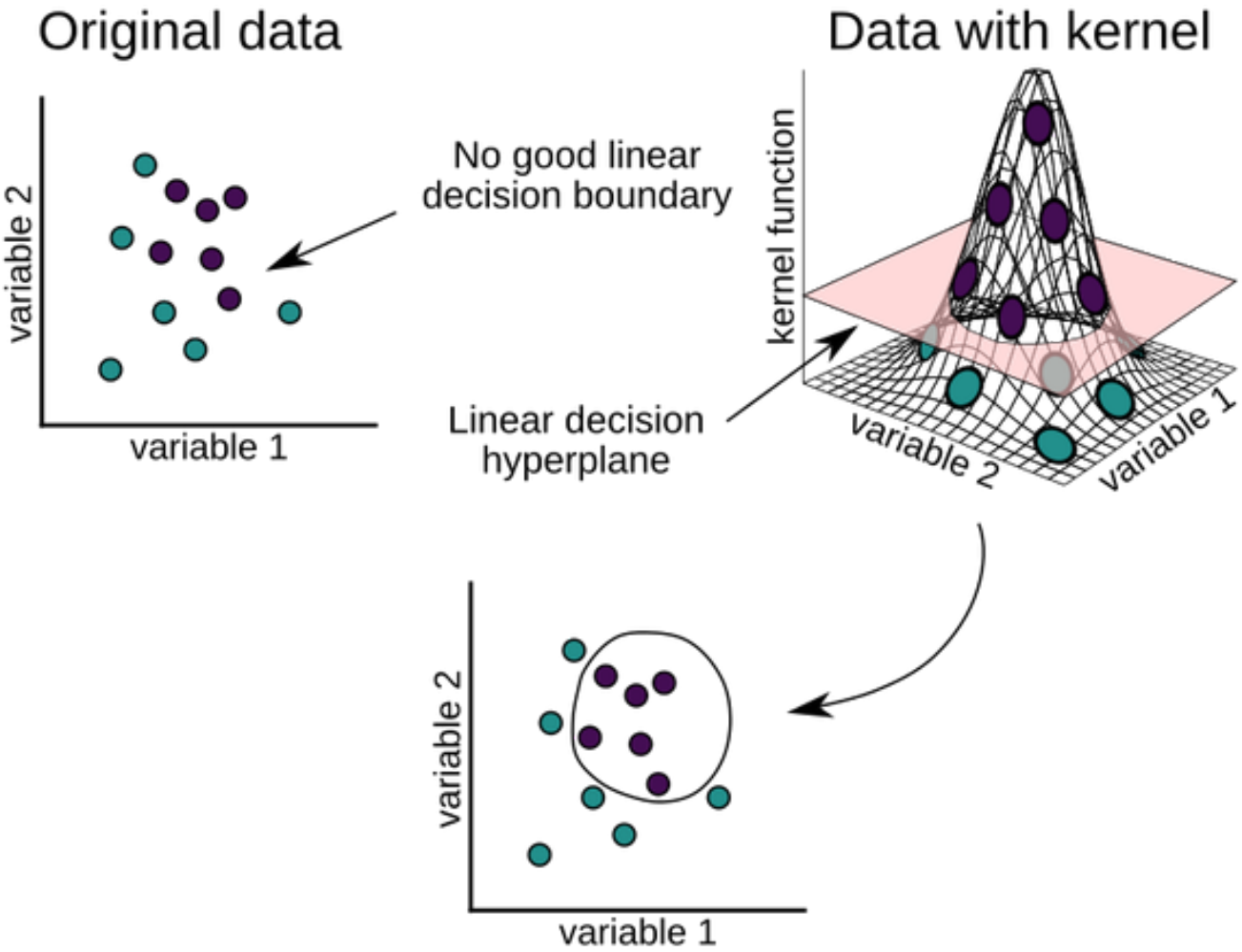


Image : https ://www.r-bloggers.com/2019/10/support-vector-machines-with-the-mlr-package/

Noyau

Pour traiter les données non linéairement séparables :

- on utilise une transformation non linéaire Φ.
- Φ permet de passer dans un espace de dimension supérieure.
- On appelle noyau K : $K(X_i, X_j) = \Phi(X_i) \bullet \Phi(X_j)$.
- Si on choisit bien Φ, le noyau s'exprime sans Φ.
- On a : $f(X) = \sum_{i=1}^{s} \lambda_i z_i K(X_i, X)$.

Exemple de Noyau

Exemple en dimension 2 :

- $X = (x_1, x_2)$
- $\Phi(X) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$
- $\Phi(X) \bullet \Phi(X') =$

L'objectif est d'exprimer $f(\Phi(X))$ en fonction de X mais sans faire intervenir Φ.
On obtient ainsi une fonction de séparation non linéaire.

Forme finale

Version duale :

Maximiser : $L(\lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \lambda_i \lambda_j z_i z_j K(X_i, X_j)$

Sous les contraintes :

$\forall i \in [\![1; n]\!]\ C \geqslant \lambda_i \geqslant 0$

$\sum_{i=1}^{n} \lambda_i z_i = 0.$

Exemples de Noyau

Quelques exemples de noyaux parmi les plus populaires :

- Linéaire $K(X_i, X_j) = X_i \bullet X_j$
  utilisé en text mining
- Polynomial : $K(X_i, X_j) = (\gamma X_i \bullet X_j + c)^d$
  utilisé en traitement de l'image
- Gaussien : $K(X_i, X_j) = e^{-\frac{\|X_i - X_j\|^2}{2\sigma^2}}$
- Radial Gaussien : $K(X_i, X_j) = e^{-\gamma \|X_i - X_j\|^2}$
  le plus courant
- Radial Laplacien : $K(X_i, X_j) = e^{-\gamma \|X_i - X_j\|}$
- Sigmoïdal : $K(X_i, X_j) = \tanh(\gamma X_i \bullet X_j + \theta)$

Exemples de Noyau

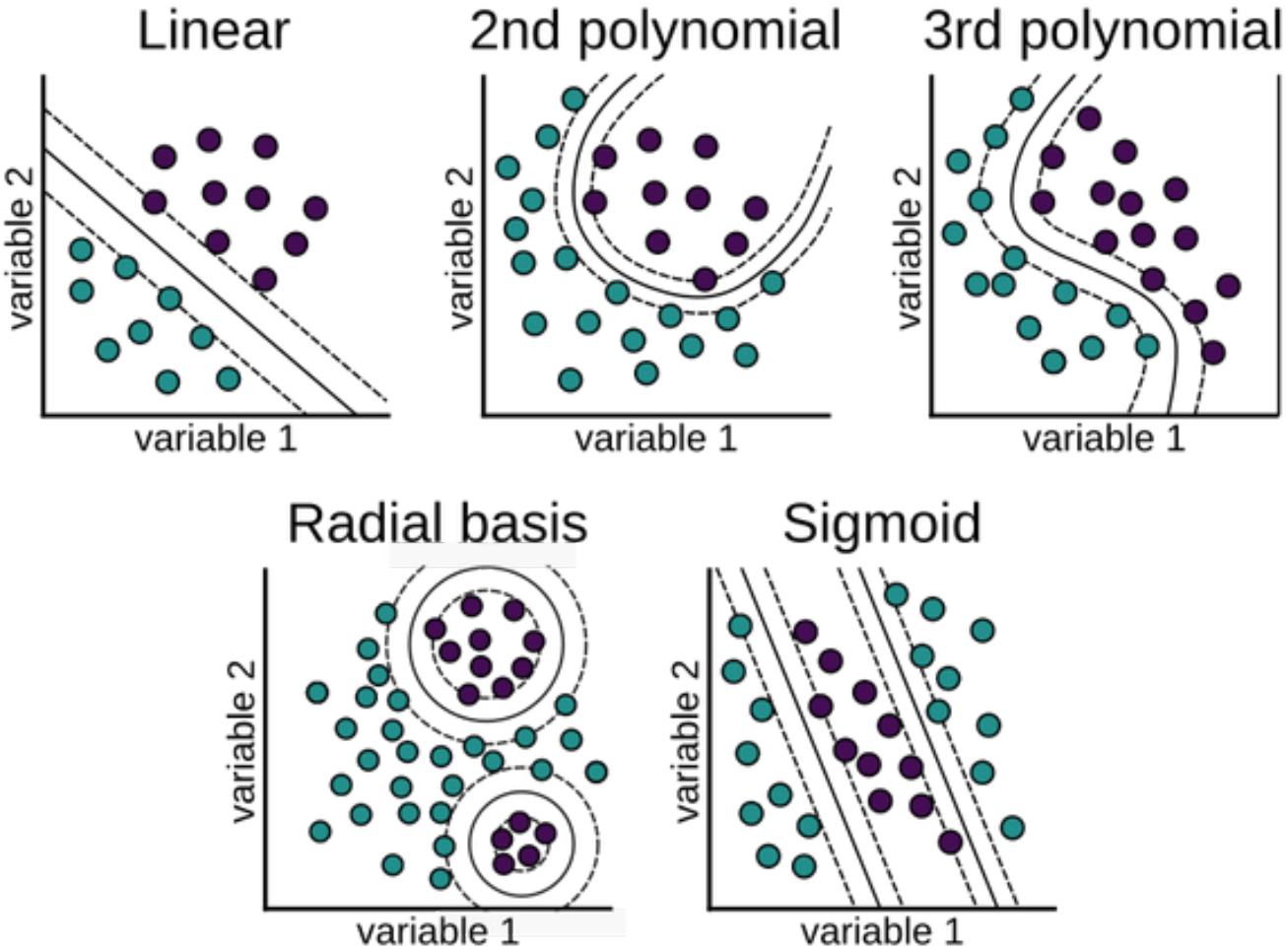


Image : https ://www.r-bloggers.com/2019/10/support-vector-machines-with-the-mlr-package/

Exemple 2

Exemple 2

Avantages

Avantages :

- Capacité à modéliser des phénomènes non linéaires. (noyaux)
- Capacité à traiter de grandes dimensions.
- Robustesse vis à vis des points aberrants. (vecteurs supports et relaxation)
- Risque plus faible de surapprentissage. (C)

Inconvénients :

- Modèles complexes.
- Sensibilité au choix des paramètres du noyau.
- Temps de calcul sur de gros volumes de données
- Généralisation à des variables multi-classes.

Rappels ?

Multiplicateur de Lagrange

- Technique utilisée en optimisation sous contraintes.
- On cherche à optimiser φ sous la contrainte $\psi(x) = 0$.
- On introduit $L(x, \lambda) = \varphi(x) + \lambda \bullet \psi(x)$.
- On montre que L optimal en x_0 si $\exists \lambda_0$ tel que $DL(x_0, \lambda_0) = 0$.
- λ_0 est appelé le multiplicateur de Lagrange.

◂ Cours

Multiplicateurs de Lagrange

Dérivées partielles :

$$\frac{\partial L}{\partial a_1}(a_1, a_2, b, \lambda) = a_1 - \sum_{i=1}^{n} \lambda_i z_i x_i$$

$$\frac{\partial L}{\partial a_2}(a_1, a_2, b, \lambda) = a_2 - \sum_{i=1}^{n} \lambda_i z_i y_i$$

$$\frac{\partial L}{\partial b}(a_1, a_2, b, \lambda) = -\sum_{i=1}^{n} \lambda_i z_i$$

$$\forall i \in [\![1; n]\!] \quad \frac{\partial L}{\partial \lambda_i}(a_1, a_2, b, \lambda) = 1 - z_i \left(a_1 x_i + a_2 y_i + b\right)$$

Multiplicateurs de Lagrange

$$\begin{cases} a_1 = \sum_{i=1}^{n} \lambda_i z_i x_i \\ a_2 = \sum_{i=1}^{n} \lambda_i z_i y_i \\ \sum_{i=1}^{n} \lambda_i z_i = 0 \\ \forall i \in [\![1; n]\!] \; 1 - z_i \left(a_1 x_i + a_2 y_i + b\right) = 0 \end{cases}$$

Multiplicateurs de Lagrange

$$\begin{cases} w_1 = \sum_{i=1}^{n} \lambda_i z_i x_i \\ w_2 = \sum_{i=1}^{n} \lambda_i z_i y_i \\ \sum_{i=1}^{n} \lambda_i z_i = 0 \\ \forall i \in [\![1; n]\!] \; 1 - z_i \left(a_1 x_i + a_2 y_i + b\right) = 0 \end{cases}$$

$$L(\lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j z_i z_j x_i x_j - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j z_i z_j y_i y_j$$

Multiplicateurs de Lagrange

$$\begin{cases} W = \sum_{i=1}^{n} \lambda_i z_i X \\ \sum_{i=1}^{n} \lambda_i z_i = 0 \\ \forall i \in [\![1; n]\!] \; 1 - z_i \left(a_1 x_i + a_2 y_i + b\right) = 0 \end{cases}$$

$$L(\lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j z_i z_j \left(X_i \bullet X_j\right)$$

◂ Cours