

## Objectifs et enjeux de l'apprentissage

- Motivation :
- explosion des capacités de stockage.
  - bases de données massives
  - données omni-présentes
  - approches génériques et automatisables

- Types de données :
- matrices de données structurées
  - séries temporelles
  - les données financières
  - l'imagerie médicale
  - les données d'internet et les logs
  - séquençage du génome humain
  - données sur les espèces / l'environnement
  - données graph sur des réseaux, des jeux ...
- ⇒ • vecteurs / matrices  
• chaînes de caractères  
• graphes / réseaux  
• pixels / séries temporelles

Les questions en machine learning :

- prédiction
- segmentation / clustering
- détection d'anomalies
- réduction de la dimension
- sélection de variables
- interprétation / parcimonie
- visualisation

- Outils :
- algèbre linéaire
  - modélisations aléatoire
  - probabilités / statistique
  - optimisation
  - traitement du signal.

On fait appel à l'apprentissage statistique car :

- hypothèse des problèmes
- No free lunch.
- choix des critères de performance
- notion de risque
- contrôle de la complexité
- validation des règles de décision
- rôle du ré-échantillonnage
- monitoring des modèles de prévision
- intégration des contraintes computatiomnelles.

Le ML, c'est plus que des stats::

- traitement de données massives, complexes, de grande dimension
- diversité des contextes : supervisé, non-supervisé ...
- couplage des principes inferentiels avec des algorithmes

## I. Introduction

### 1. Modèle statistique

Observation comme réalisations de  $X$  variable aléatoire de loi inconnue  $P_\theta$ .

$$X_1, \dots, X_n \text{ iid. } X \sim P_\theta$$

On suppose  $X$  à valeurs dans  $(E, \mathbb{E})$ .

Un modèle statistique est un triplet  $\mathcal{M} : (E, \mathbb{E}, P)$  où  $P = \{P_\theta : \theta \in \Theta\}$  famille de lois candidates pour  $P^*$ .

$\Theta$  est un paramétrage de  $P$ , on note  $P^\theta = P_{\theta^\star}$ .

Le modèle est paramétrique si  $\Theta$  est un sous-espace vectoriel d'un espace euclidien.  
Le modèle est dit non-paramétrique autrement.

ex: Soit  $X = \begin{cases} 1 & \text{si pile} \\ 0 & \text{si face} \end{cases}$

$$p = P\{X = 1\}$$
$$\theta = \mathbb{E}(X) = p \cdot 1 + 0 \cdot (1-p) = p.$$
$$\text{Var}(X) = \mathbb{E}((X - \mathbb{E}(X))^2) = p(1-p)$$

$$P_p \{X_1 = i_1, \dots, X_n = i_n\} = P_p \{X_1 = i_1\} \cdot \dots \cdot P_p \{X_n = i_n\}$$

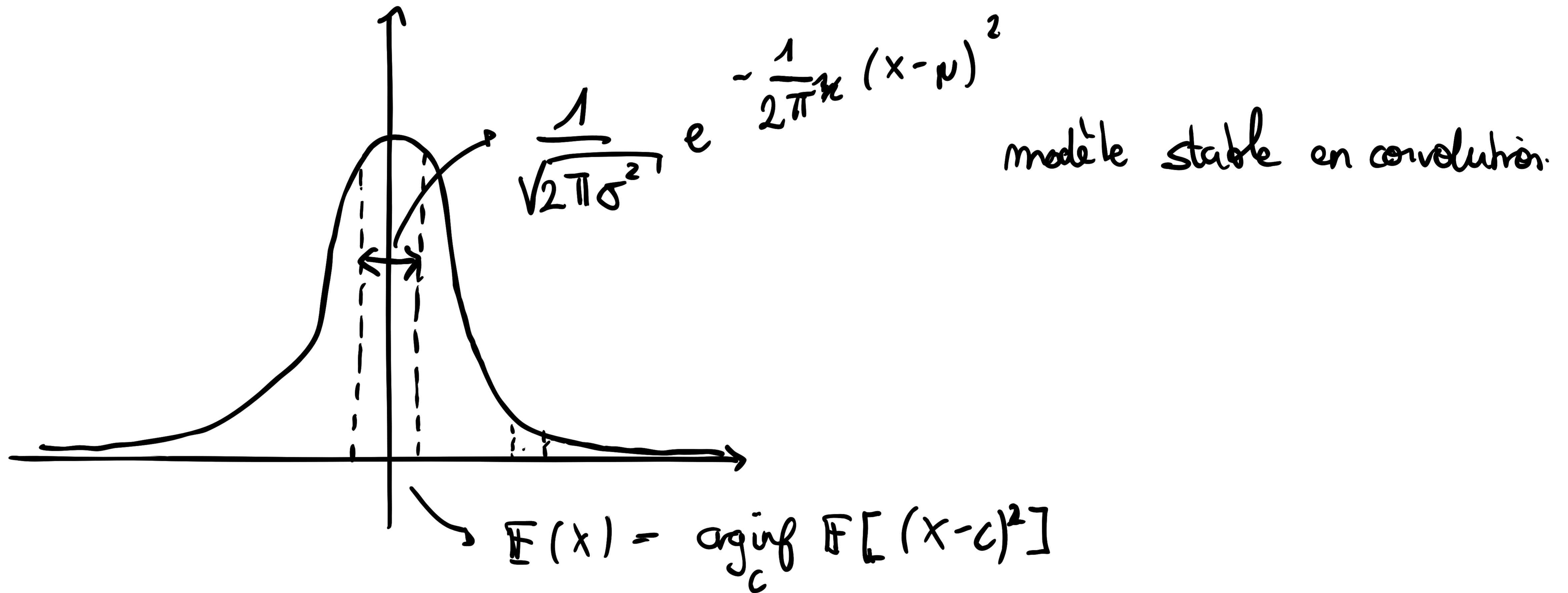
• Cela nous permet de définir la vraisemblance:

$$P_p \{ X_1 = i_1, \dots, X_n = i_n \} = p^{\sum_{i=1}^n \mathbb{1}_{\{X_i=i\}}} (1-p)^{\sum_{i=1}^n \mathbb{1}_{\{X_i \neq i\}}} = L_n(p)$$

• La log-vraisemblance est:  $\ln(p) = \log L_n(p) = S_n \log p + (1-S_n) \log (1-p)$   
 où  $S_n = \sum_{i=1}^n \mathbb{1}_{\{X_i=1\}} = \sum_{i=1}^n \mathbb{1}_{\{X_i=1\}}$ .  
 On trouve le maximum pour notre paramètre avec  $\ln'(p) = 0$

$$\text{L} \cdot \ln(p) = p^{S_n} (1-p)^{n-S_n} \Rightarrow \ln'(p) = 0 \Rightarrow \hat{p} = \frac{S_n}{n} = \frac{X_1 + \dots + X_n}{n}$$

Dans le cas normal:



$$\begin{aligned} \text{Risque quadratique: } R(\hat{\theta}_n, \theta^*) &= E_{\theta^*} ((\hat{\theta}_n - \theta^*)^2) \\ &= (\mathbb{E}(\hat{\theta}_n) - \theta^*)^2 + V_{\theta^*}(\hat{\theta}_n). \end{aligned}$$

↑  
biais      ↑  
variance

• Régression:  $Y = X\beta + \varepsilon$

$$\left\{ \begin{array}{l} Y \in \mathbb{R}^n \\ X \in \mathbb{R}^{n \times p} \rightarrow X^T X \text{ inversible} \\ \beta \in \mathbb{R}^p \\ \varepsilon \sim N_n(0, \sigma^2 I_n) \end{array} \right.$$

$$\Rightarrow \hat{\beta} = (X^T X)^{-1} X^T Y$$

$$\hat{s}^2 = \frac{1}{n} \|Y - X\hat{\beta}\|^2$$

## 2. les problèmes statistiques revisités.

### Supervisé

Couple de v.a =  $(X, Y) \sim P_{\text{inconnue}}$ .

- $X$  = vecteur d'entrée à valeurs dans  $\mathcal{X}(\mathbb{R}^d)$ ,  $d \geq 1$ .
- $Y$  = label / étiquette dans  $y \in \mathbb{R}$
- Règle prédictive :  $g(x) = \beta x + \alpha$  si prédicteur linéaire,  $g: \mathcal{X} \rightarrow \mathcal{Y}$  généralement.

Fonction de perte :  $l: \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+$ .

Risque inconnu = Erreur de généralisation =  $L(g) = \mathbb{E}(l(Y, g(x)))$  à minimiser sur  $g \in \mathcal{G}$ .

- Données =  $D_n = \{(X_1, Y_1), \dots, (X_n, Y_n)\} \stackrel{iid}{\sim} P$ .

ex 1 : Régression

Fonction de perte = erreur quadratique  $l(y, z) = (y - z)^2$

Solution optimale :  $g^*(x) = \mathbb{E}(Y | X=x)$

ex 2 : Scoring

Données de classification  $\mathcal{Y} = \{0, 1\}$

Préba à postériori :  $\eta(x) = \mathbb{E}(Y | X=x) = P(Y=1 | X=x)$

Régression logistique :  $f(x) = \log(\eta(x) / (1-\eta(x)))$

Régression logistique linéaire :  $f(x) = \beta x + \alpha$

ex 3 : Classification binaire

Prédicteur de l'état d'un système :  $\mathcal{Y} = \{-1, +1\}$

Fonction de perte :  $l(y, z) = \mathbb{1}\{y \neq z\}$

Risque d'erreur :  $L(g) = P(Y \neq g(x)) = P(Y \cdot g(x) < 0) = \mathbb{E}(\mathbb{1}_{Y<0}(-Y \cdot g(x)))$

ex 4 : Classification multi-classe

$\mathcal{Y} = \{1, \dots, M\}$ . Fonction de perte :  $l(y, z) = \mathbb{1}\{y \neq z\}$

ex 5 : Régression ordinaire

$\mathcal{Y} = \{1, \dots, M\}$  label ordinal

Fonction de perte  $l(y, z) = (y - z)^2$

### Non-supervisé

Pas d'étiquette  $Y$ .

Modèle statistique non-paramétrique :  $\{\rho(x, \theta) : \theta \in \Theta\}$

Reconnaître la densité  $f(x)$  à partir de  $D_n = \{X_1, \dots, X_n\}$ .

Fonction de perte :  $l(x, \theta) = -\log \rho(x, \theta)$ .

Applications : clustering, modes, détection d'anomalies

Ex 7: Rang et score comme de la classification binaire.

$$\eta(x) = P(Y=1 | X=x).$$

Ordonnement:  $s: \mathcal{X} \rightarrow \mathbb{R}$

Trouver  $s$  qui ordonne les éléments de  $\mathcal{X}$  comme  $\eta$

↳ Vrais positifs:  $TPR_s(t) = P(s(x) \geq t | Y=1)$

Faux positifs:  $FPR_s(t) = P(s(x) \geq t | Y=-1)$

$$ROC: t \mapsto (FPR_s(t), TPR_s(t))$$

### 3. Principes statistiques et algorithmes.

Paradigme de l'apprentissage à travers l'exemple de la classification.

$$(X, Y) \sim P_{\text{inconnue}}$$

$X$  v.a d'entrée

$Y$  label binaire à valeurs dans  $y = \{-1, +1\}$

À partir d'exemples:  $(x_1, y_1), \dots, (x_n, y_n)$ , construire un classifieur  $C: x \in \mathcal{X} \mapsto C(x) \in \{-1, +1\}$  appartenant à une classe  $\mathcal{G}$  de risque minimum:

$$L(C) = \mathbb{E}(1_{\{Y \neq C(x)\}})$$

où  $\mathcal{G}$  est en correspondance bi-univoque avec la classe:

$$\{ \{x \in \mathcal{X}: C(x) = +1\} : C \in \mathcal{G} \}$$

⚠ Apprentissage  $\neq$  modélisation.

Idealement, calculer  $C^* = \arg \min_{C \in \mathcal{G}} L(C)$ .

$L$  et  $P$  inconnus.

$n(x) = P(Y=+1 | X=x)$  probabilité à posteriori.

On pose  $p = P(Y=+1)$ .

Minimiser  $\mathbb{E}(1_{\{Y \neq g(x)\}} | x)$  montre que:

$$C^*(x) = 2 \cdot 1_{\{n(x) > 1/2\}} - 1, \quad x \in \mathcal{X}.$$

On prédit le label le plus probable au vu de l'observation  $X=x$ , classifieur de Bayes.

↳ Risque minimum théorique:  $L^* = L(C^*) = 1/2 - \mathbb{E}[|n(x) - 1/2|]$

↳ La distribution de  $n(x)$  autour de  $1/2$  régit la difficulté du problème.

On veut minimiser le risque empirique:

Données d'apprentissage:  $D_n = \{(x_1, y_1), \dots, (x_n, y_n)\}$

Candidat  $C: X \rightarrow \{-1, 1\}$  à la classe  $\mathcal{G}$

Risque empirique = Erreur d'apprentissage

$$\hat{L}_n(C) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{y_i \neq C(x_i)\} \text{ à minimiser sur la classe } \mathcal{G}.$$

Solution: minimiseur du risque empirique:  $\hat{C}_n = \underset{C \in \mathcal{G}}{\operatorname{arg\,min}} \hat{L}_n(C)$ .

L'apprentissage fonctionne si  $\hat{L}_n(C)$  proche de  $L(C)$  lorsque  $C$  degreit  $\mathcal{G}$ .  
 $\hat{L}_n(C) \xrightarrow{n \rightarrow \infty} L(C)$

Théorie de Vapnik-Chervonenkis: garanties vis à vis de la prédition si  $\mathcal{G}$  pas trop complexe.  $\Rightarrow$  ! Overfitting

Sous quelles conditions l'apprentissage fonctionne? Et donc  $L(\hat{C}_n)$  proche de  $L^*$ ?

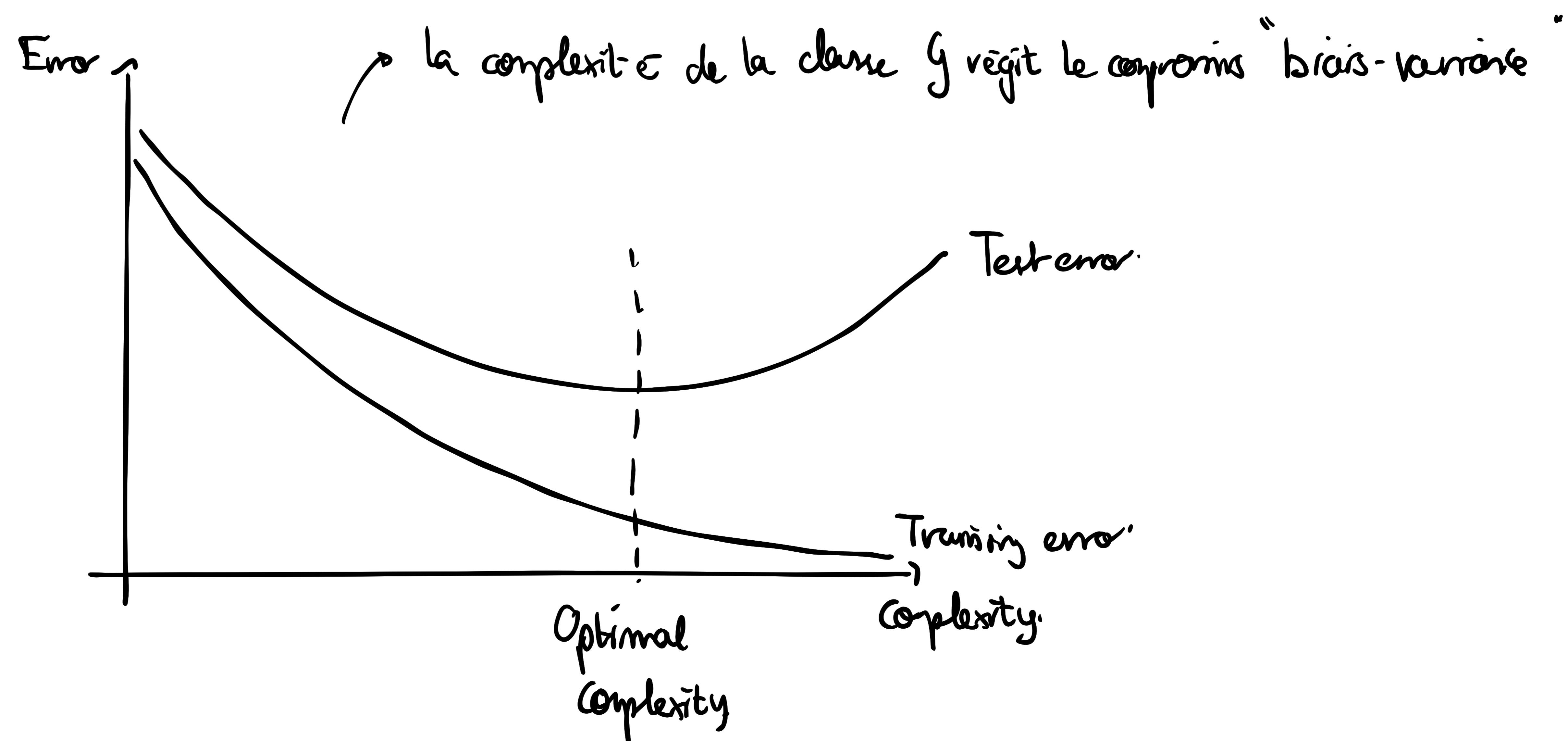
Décomposition biais-variance de l'excès de risque:

$$L(\hat{C}_n) - L^* \leq 2 \underbrace{\sup_{C \in \mathcal{G}} |L(C) - \hat{L}_n(C)|}_{\text{Biais}} + \underbrace{\inf_{C \in \mathcal{G}} (C) - L^*}_{\text{Varianc}}$$

Contrôlé au moyen des résultats de concentration

$$\text{pour } Z = \{L(C) - \hat{L}_n(C)\}_{C \in \mathcal{G}}$$

Dépend de la classe  $\mathcal{G}$ , pas des données



### • Théorie de Vapnik et Chervonenkis:

Permet de contrôler l'erreur de généralisation.

↳ Règles linéaires / quadratiques / paramétriques. (LDA, QDA, SVM, RN)

↳ Règles décrites par des arbres de décision.

La minimisation du risque empirique (ERM) est NP-difficile:

↳ on ne peut pas conclure sans avoir exploré toutes les données.

↳  $\min_{f \in \mathcal{G}} L_n(f)$  à optimiser:

- Optimisation continue : descente de gradient

- Optimisation discrète : enumerer intelligemment.

→ Stratégies gloutonnes.

En pratique :  $\text{sg}(x) = \text{sign}(f(x))$ ,  $L(f) = \mathbb{E}[1\{\cdot - f(x) > 0\}]$ .

↳ On remplace la perte  $l(u) = 1\{u > 0\}$  par une version régulière  $\tilde{l}(u)$

→ Risque  $\tilde{L}_n(f) = \mathbb{E}[\tilde{l}(\cdot - f(x))]$ :

- SVM  $\tilde{l}(u) = \max(0, 1+u)$

- Boosting  $\tilde{l}(u) = \exp(u)$

- Réseaux de neurones  $\tilde{l}(u) = \tanh(u)$

Approximation stochastique:

Minimiser une version lissée (convexifiée) et éventuellement pénalisée du risque empirique :  $\min_{f \in \mathcal{G}} \tilde{L}_n(f)$ :

$$\tilde{L}_n(f) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n \tilde{l}(-y_i f(x_i)) + \text{pen}(f).$$

En général, approximation stochastique itérative:  $f_{t+1} = f_t - \rho_t \nabla f \tilde{L}_n(f_t)$

↳ Logit, Neural Networks, Linear SVM... sont basés dessus.

• Perceptron monocouche (Rosenblatt, 1962):

(cas binaire :  $Y \in \{-1, +1\} \Rightarrow g(x) = \text{sign}(a + \langle b, x \rangle)$ ,  $\tilde{l}(u) = u$ )

Itérations :

- choisir une observation  $(X_i, Y_i)$  parmi les observations mal classées par la règle courante

$$- (a, b) \leftarrow (a, b) + \rho(Y_i, y_i x_i)$$

↳ Descente de gradient stochastique.

## 4. Evaluation du risque et sélection de modèle

- Régler le bon niveau de complexité
- Erreur d'apprentissage et de généralisation:  
Apprentissage depuis:  $D_n = \{(x_1, y_1), \dots, (x_n, y_n)\}$ .
- Classifieur  $\hat{C}_n \in \mathcal{G}$  construit à partir d'une méthode réalisant la minimisation du risque empirique. Aléatoire, dépend de  $D_n$ .  
Erreur :  $L(\hat{C}_n) = \mathbb{E}[\mathbb{1}\{Y \neq \hat{C}_n(X)\} | D_n]$ 
  - ↳ l'espérance est prise sur un couple  $(X, Y)$  indépendant de  $D_n$

Mais l'erreur d'apprentissage n'est pas un bon estimateur de l'erreur.

Elle se réduit à 0 quand  $\mathcal{G}$  suffisamment complexe. → Surajustement / Overfitting .

$$\hat{L}_n(\hat{C}_n) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{Y_i \neq \hat{C}_n(x_i)\}$$

Objectif de sélection du modèle et évaluation du risque.

- Big data:
  - ↳ Données divisées en 3:
    - Apprentissage → 50%
    - Validation → 25%
    - Test → 25%
  - ↳  $k > 1$  modèles candidats :  $G_1, \dots, G_k$ .
    - Pour chaque modèle, appliquer ERM aux données d'apprentissage.
    - Utiliser les données de validation pour trouver le  $\hat{k}$  optimal  $\in \{1, \dots, K\}$ .
    - Estimer son erreur avec les données test.
  - ↳ Possibilité de réaliser l'apprentissage sur la quasi-totalité des données par re-sampling.
- Validation croisée:  $K$ -fold (1,5 ou 10). ou  $K=n$  ("leave one out")  
Diviser aléatoirement les données en  $K$  parties égales.  
Pour tout  $k \in \{1, \dots, K\}$ .
  - apprendre  $\hat{C}^{(-k)}$  à partir de toutes les données sauf celles de la  $k^{\text{ème}}$  partie.
  - calculer l'erreur réalisée par  $\hat{C}^{(-k)}$  sur les données de la  $k^{\text{ème}}$  partie.
- ⇒ Puis moyenne les  $K$  quantités

- Bootstrap: remplacer la distribution inconnue des données par la distribution empirique.  
Par approximation de Monte-Carlo, réitérer  $B$  fois et majorer les erreurs.

La Application: KNearest Neighbors.

Soit  $K \geq 1$ . Sur  $\mathbb{R}^p$ , on considère une métrique  $d$  (ex: distance euclidienne).

$\forall x$ , soit  $\sigma = \sigma_x$ , permutation de  $\{1, \dots, n\}$  tq:

$$d(x, x_{\sigma(1)}) \leq \dots \leq d(x, x_{\sigma(n)}).$$

On considère les  $K$  plus proches voisins:  $\{x_{\sigma(1)}, \dots, x_{\sigma(K)}\}$

la Vote majoritaire:  $N_y = \text{Card} \{ k \in \{1, \dots, K\} : y_{\sigma(k)} = y \}$ ,  $y \in \{-1, 1\}$ .

$$C(x) = \arg \max_{y \in \{-1, +1\}} N_y.$$

↳ Si  $K=1 \rightarrow$  sur-ajustement

Si  $K=n \rightarrow$  sous-ajustement.

↳ Choisir de  $K$  par plan d'expérience, validation croisée, bootstrap.

## Premiers Algorithmes

• Classification binaire : le Classifieur de Bayes.

Label  $Y \in \{-1, +1\}$

Entrée :  $x \in \mathbb{R}^d$ ,  $d \gg 1$ .

$g$  booléenne :  $\mathbb{R}^d \rightarrow \{-1, +1\}$

$x \mapsto g(x)$ .

$$g(x) = 2 \mathbb{1}\{z : g(z) = +1\} - 1$$

$$\begin{aligned} \text{Objectif : minimisation de } L(g) &= P(Y \neq g(x)) = \mathbb{E}(1\{Y \neq g(x)\}) \\ &= \mathbb{E}[1\{Y_h(x) > 0\}] \end{aligned}$$

tout en sachant que la loi de  $(X, Y)$  est inconnue.

Quelle est la solution optimale / idéale ?

$$\hookrightarrow L(g) = \mathbb{E}\left[\left[1\{Y=+1\} \mathbb{1}\{g(x) = -1\} + [1\{Y=-1\} \mathbb{1}\{g(x) = +1\}]\right] | X\right]$$

$\underbrace{\hspace{10em}}$

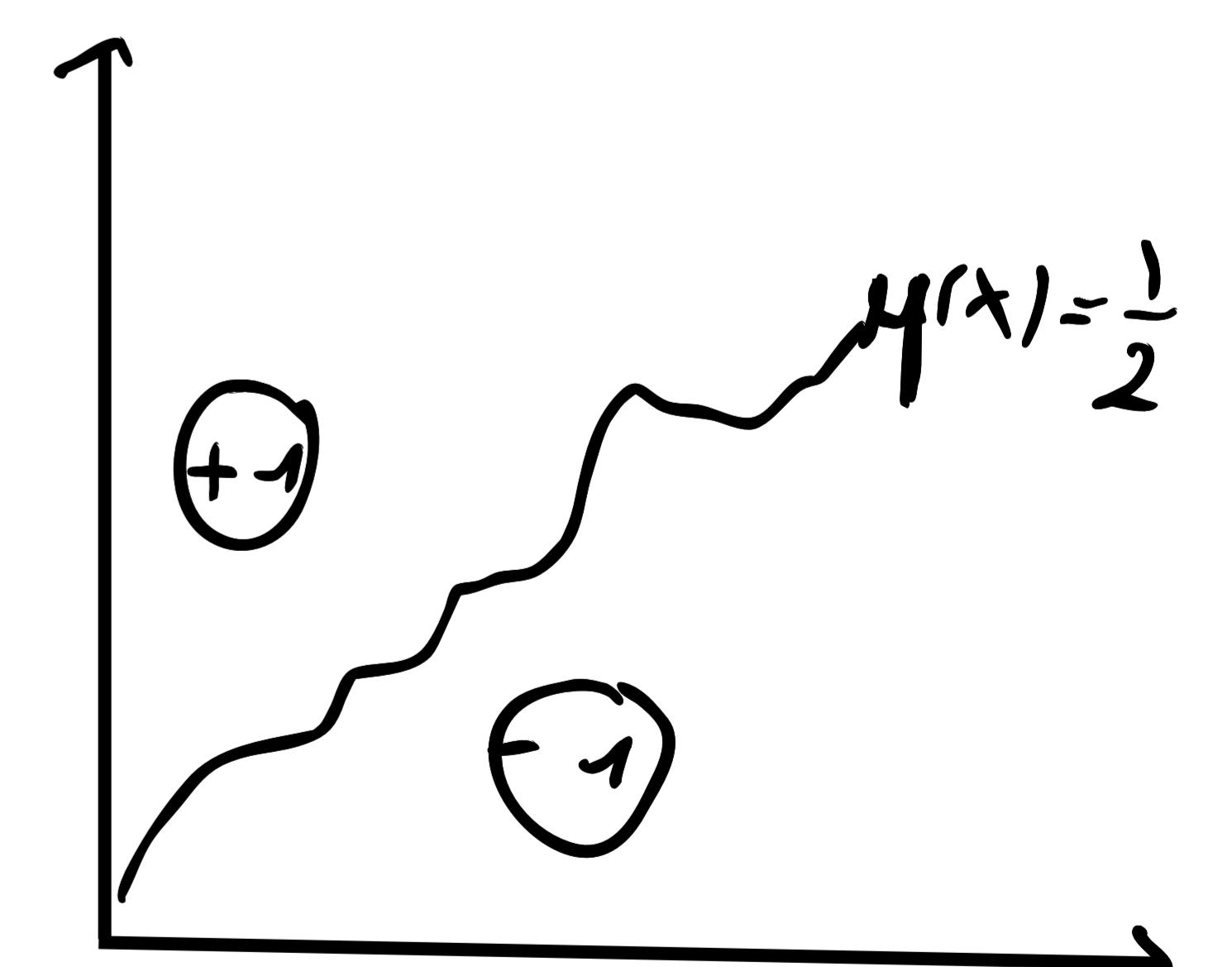
Deux manières de se tromper.

$$\begin{aligned} &= \mathbb{E}[1\{g(x) = -1\} \mathbb{E}[1\{Y=+1\}|X] + 1\{g(x) = +1\} \mathbb{E}[1\{Y=-1\}|X]] \text{ par Fubini.} \\ \text{La probabilité à postérieur est: } &y(x) = P(Y=1|x) = 1 - P(Y=-1|x) = \mathbb{E}[1\{Y=1\}|x] \\ \Rightarrow &= \mathbb{E}[y(x) \mathbb{1}\{g(x) = -1\} + (1-y(x)) \mathbb{1}\{g(x) = +1\}] \end{aligned}$$

À  $x$  fixé, le minimum pour  $g(x) = +1$  si  $|1-y(x)| < y(x) \Leftrightarrow y(x) \geq \frac{1}{2}$   
 $\hookrightarrow = -1$  sinon

Ainsi, le minimum de  $L(g)$  est atteint pour:

$$g^*(x) = 2 \mathbb{1}\{y(x) \geq \frac{1}{2}\} - 1.$$



But : Reconnaître statistiquement  $\{x : y(x) \geq \frac{1}{2}\}$ .

Plus simple que d'estimer par Bayes :  $y(x) = y(x^{(1)}, \dots, x^{(d)})$

à cause notamment du fléau de la dimension (curse of dimensionality).

On ne connaît jamais  $g^*(x)$  et on ne peut jamais connaître la frontière de décision si non-simulées.

→ Solution : Perceptron monocouche.

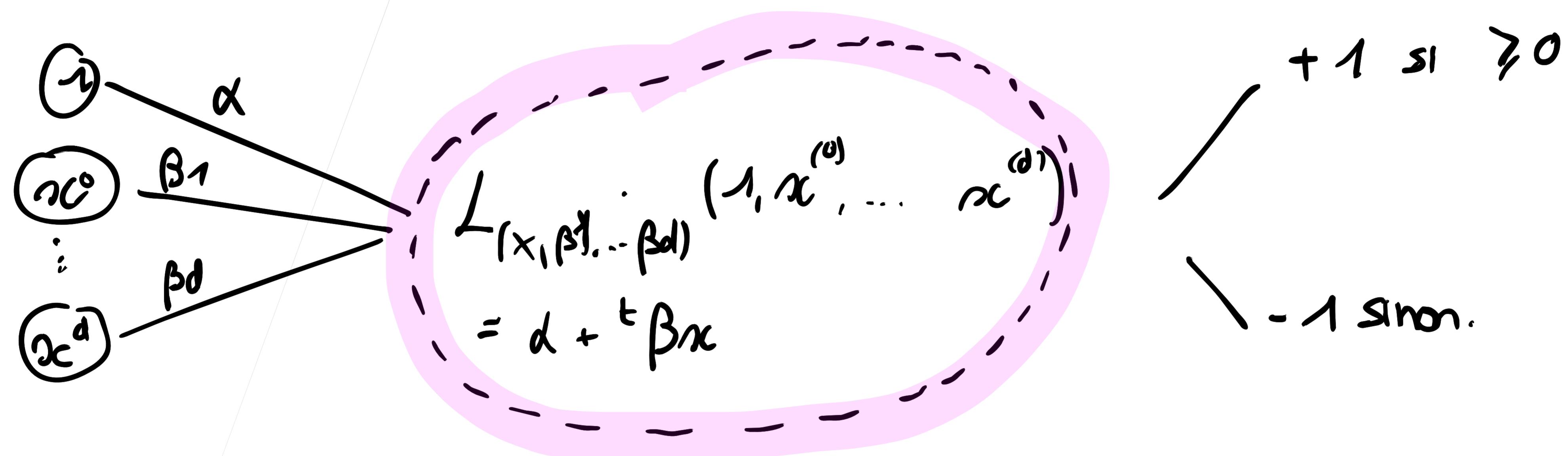
• Perceptron monocouche : (F. Rosenblatt)

Règle linéaire / affine  $g(x) = \text{sgn}(d + {}^t \beta x)$

$$\begin{array}{c} \lambda + {}^t \beta x = 0 \\ \oplus \quad \ominus \\ \rightarrow \end{array}$$

→ NB: pas adapté à la desc. degrad.

$\min_{(\alpha, \beta)} \hat{L}_n(g)$ ,  $\hat{L}_n(g) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{-(\lambda + {}^t \beta x_i) y_i > 0\}$  fonction empirique.



Rappel sur la dérivation du minimum:

$f(x) = F'(x) = 0$ . Soit  $x^*$  la solution.

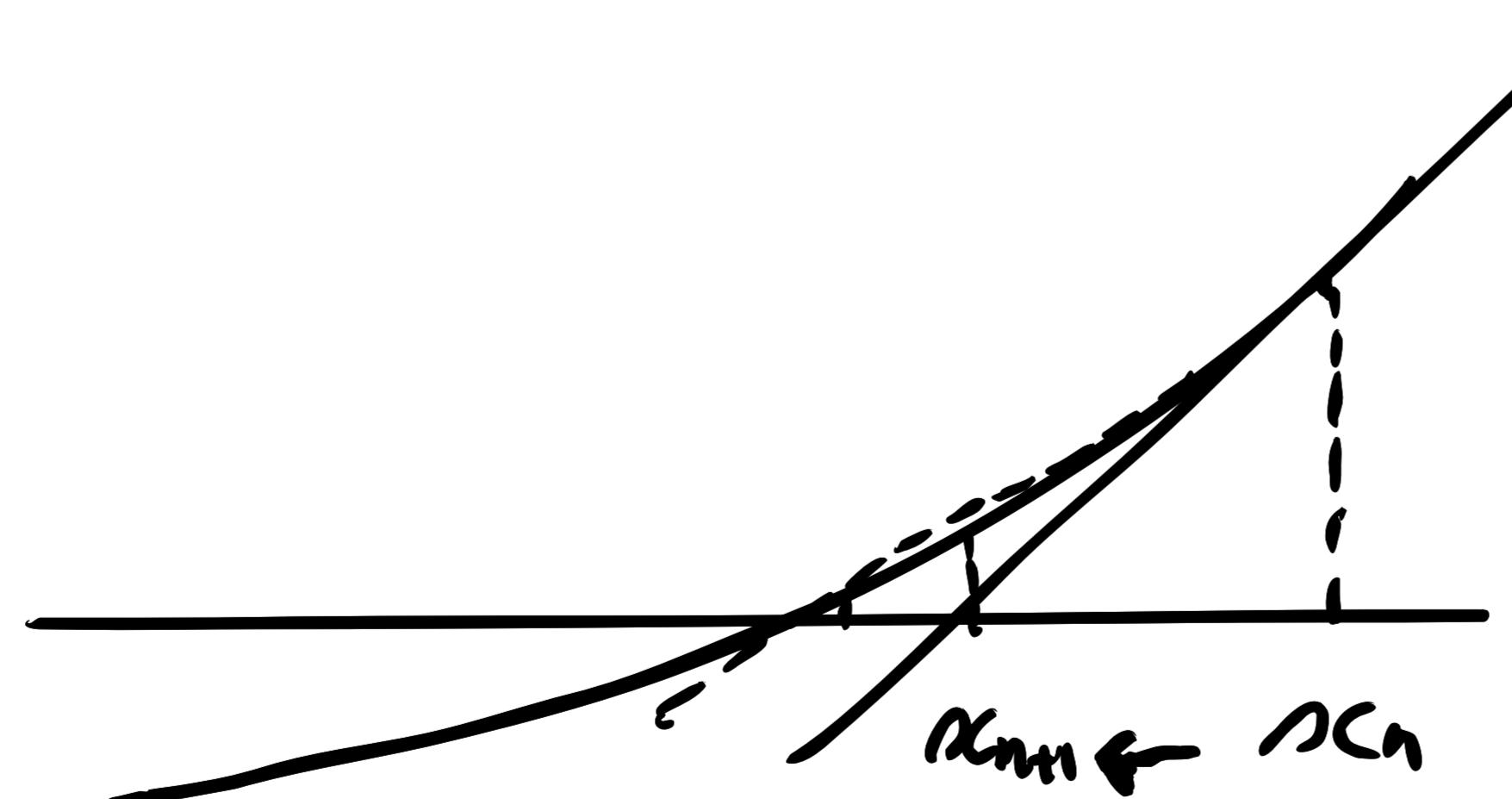
Par Taylor:  $0 = f(x^*) \approx f(x) + (x^* - x) f'(x)$

$$\Rightarrow x = x^* + \frac{f(x)}{f'(x)} \Rightarrow x^* = x - \frac{f(x)}{f'(x)}$$

La méthode de Newton nous permet d'appliquer la descente de gradient:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

La plupart du temps, il est coûteux de calculer  $f'(x_n)$  et on applique un pas.



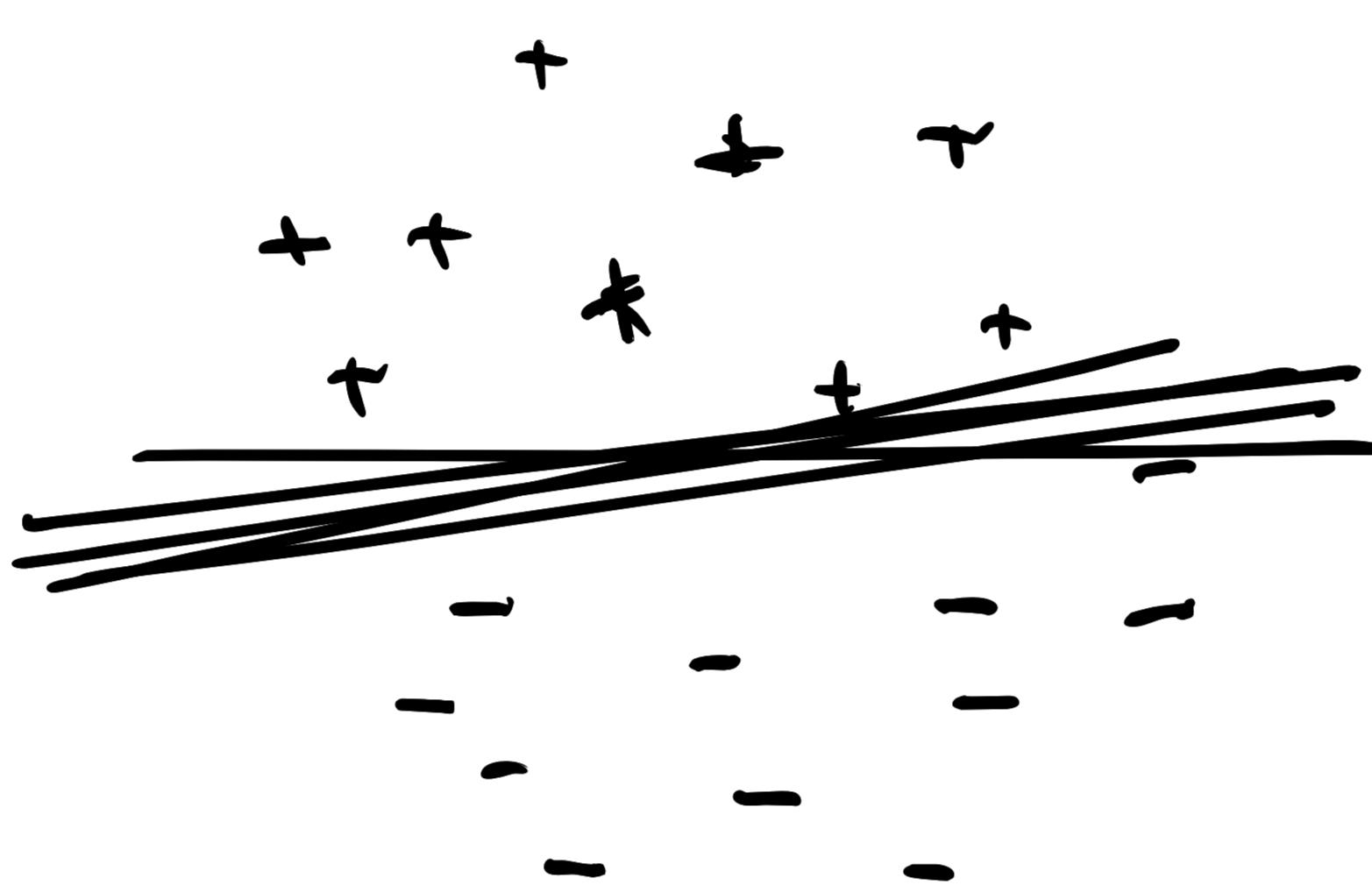
Comment appliquer ceci au perceptron ?

Si  $L$  est connue, on peut faire un lissage du risque empirique dans le cadre continu.

$$\min \sum_{i=1}^n -(\alpha + {}^t \beta x_i) y \Rightarrow \nabla_{\alpha, \beta} \sum_{i=1}^n (\alpha + {}^t \beta x_i) y = 0.$$

Mais  $L$  est inconnue. On applique donc une descente de gradient stochastique. La descente de gradient a lieu point par point seulement sur les points mal classés.

Si les données peuvent être séparées par un hyperplan, il y a une infinité d'hyperplans séparateurs. Une des faiblesses de l'algorithme n'est d'en sélectionner qu'un. Il faudrait trouver un hyperplan à vaste marge qui maximise les distances selon des marges.



Autre limite : on suppose que les données sont des données linéairement séparables. Une réponse linéaire à ce problème est la SVM (développé plus tard).

- **Régression logistique (Linéaire) :**

Repose sur l'idée du plug-in : d'après Bayes :  $g^*(x) = 2 \mathbb{1}\{y(x) > \frac{1}{2}\} - 1$

L'idée est d'estimer  $y(x)$  par  $\hat{y}(x)$ , et de "plugger" cet estimateur dans celui de Bayes :  $\hat{g}(x) = 2 \mathbb{1}\{\hat{y}(x) > \frac{1}{2}\} - 1$ .

On montrera par la suite que :  $L(\hat{g}) - L^* \leq \mathbb{E}[\hat{y}(x) - y(x)]$

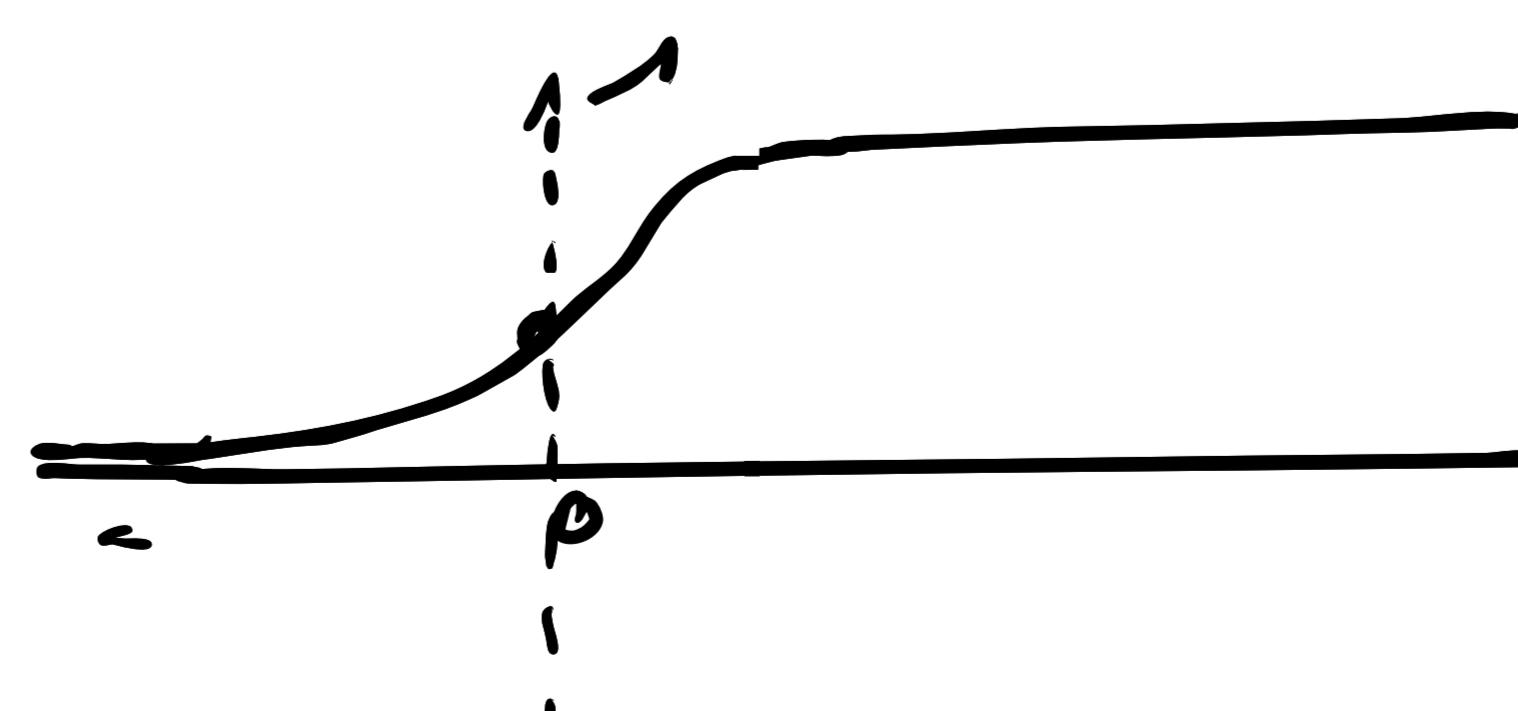
a) **Régression**:  $y \in \mathbb{R}$  :  $y = \alpha + {}^t \beta x + \varepsilon \rightarrow$  indépendant de  $x$ , qplé invari.  $\mathbb{E}(\varepsilon) = 0$ ,  $\mathbb{E}(\varepsilon|x) = \mathbb{E}(\varepsilon) = 0$ .

On suppose un modèle linéaire.

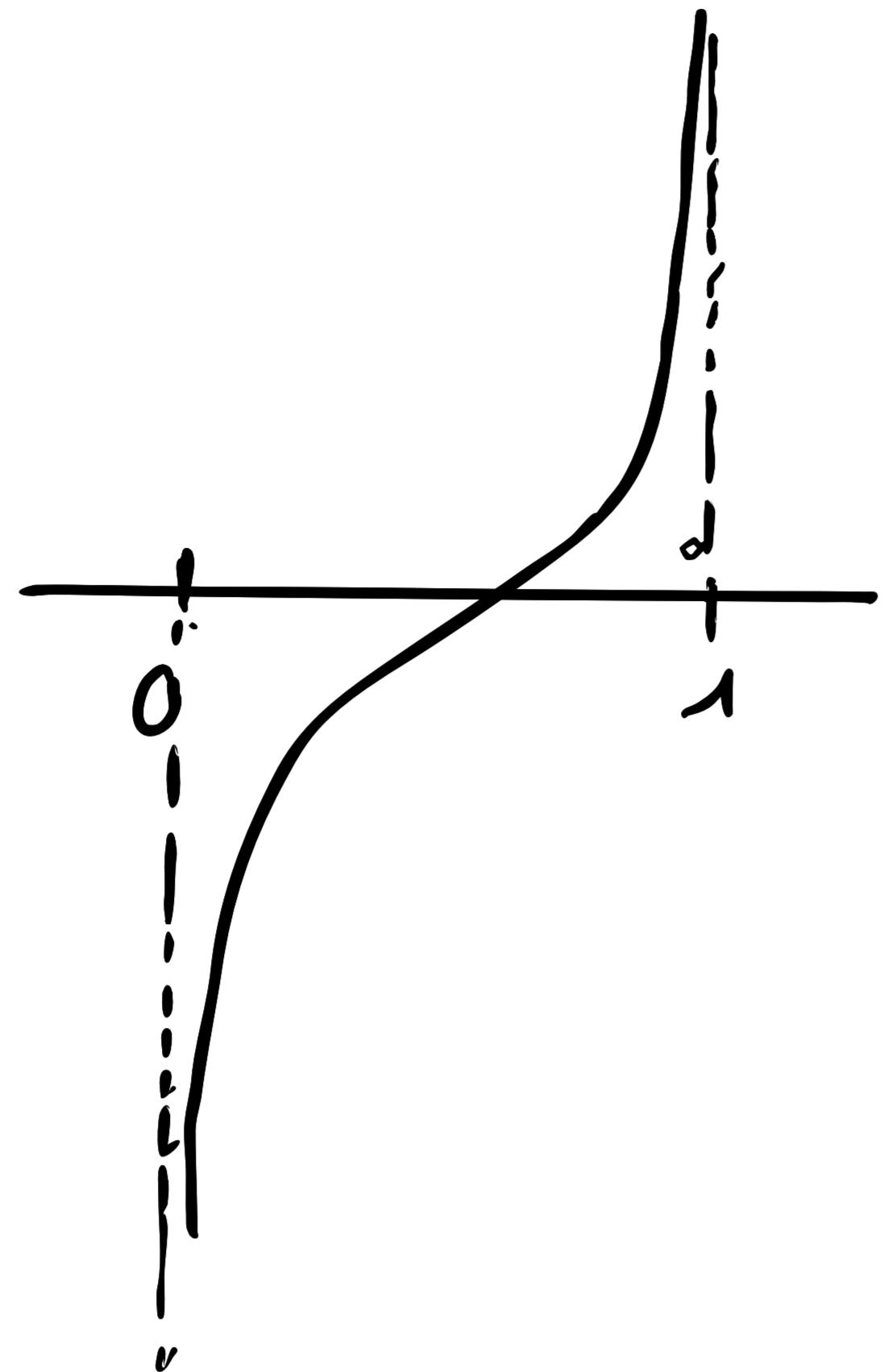
b) **Classification**:  $y \in \{-1, +1\}$

L'idée est d'envoyer  $\mathbb{R}$  dans  $[0, 1]$  grâce à la transformée logistique (logit).

$$u \in \mathbb{R} \rightarrow \frac{e^u}{1+e^u} \in [0, 1]$$



La réciproque de cette transformation est :  $\text{logit}(v) = \log\left(\frac{v}{1-v}\right)$



$$\text{logit} : ]0, 1[ \rightarrow \mathbb{R}$$

$$\text{Logit } y(x) = \alpha + \beta x \Rightarrow y_{\alpha, \beta}(x) = \frac{e^{\alpha + \beta x}}{1 + e^{\alpha + \beta x}} = y_{\alpha, \beta}(x)$$

$$y_{\alpha, \beta}(x) = \frac{e^{\alpha + \beta x}}{1 + e^{\alpha + \beta x}}$$

Il existe plein d'autres transformations pour envoier  $[0, 1]$  sur  $\mathbb{R}$  et inversement.

Par exemple :  $\Phi(u) = P(X \leq u)$ ,  $X \sim N(0, \sigma^2)$  fonction non pas logit mais probit.

Comment trouver  $\hat{\alpha}$  et  $\hat{\beta}$ ? Par maximum de vraisemblance conditionnel,  $(x_1, y_1), \dots, (x_n, y_n)$ .

$$\begin{aligned} & \ln P(\alpha, \beta) \{ Y_1 = y_1, \dots, Y_n = y_n \mid X_1 = x_1, \dots, X_n = x_n \} \\ &= P(\alpha, \beta) \underbrace{\{Y_1 = y_1 \mid X_1 = x_1\}}_{\downarrow} \times P(\alpha, \beta) \{Y_2 = y_2 \mid X_2 = x_2\} \times \dots \end{aligned} \quad ) \text{ indépendance.}$$

$$\text{Si } y_1 = 1, P = y_{\alpha, \beta}(x)$$

$$\text{Si } y_1 = -1, P = (1 - y_{\alpha, \beta}(x))$$

$$\begin{aligned} &= (y_{\alpha, \beta}(x_1))^{1\{y_1=1\}} (1 - y_{\alpha, \beta}(x_1))^{1\{y_1=-1\}} \cdot \dots \cdot (y_{\alpha, \beta}(x_n))^{1\{y_n=1\}} (1 - y_{\alpha, \beta}(x_n))^{1\{y_n=-1\}} \\ &= \Lambda_n(\alpha, \beta) \text{ la vraisemblance.} \end{aligned}$$

On s'intéresse à la log vraisemblance :  $\ln(\alpha, \beta) = \log \Lambda_n(\alpha, \beta)$ .

$\hookrightarrow \nabla \ln(\alpha, \beta) = 0$  est un système en  $(\alpha, \beta)$  qui est non linéaire donc n'offre pas de solution explicite  $\Rightarrow \neq$  regression linéaire.

On applique donc une descente de gradient Newton-Raphson. Nous donne  $\hat{\alpha}, \hat{\beta}$  numériquement.

$$\hookrightarrow (\hat{\alpha}, \hat{\beta}) \rightarrow \hat{y}_{\hat{\alpha}, \hat{\beta}}(x) \rightarrow \hat{g}(x) = 2 \mathbb{1}\{\hat{y}_{\hat{\alpha}, \hat{\beta}}(x) > \frac{1}{2}\} - 1$$

$$\Rightarrow \hat{g}(x) = 2 \mathbb{1} \left\{ \frac{e^{\hat{\alpha} + \hat{\beta} x}}{1 + e^{\hat{\alpha} + \hat{\beta} x}} > \frac{1}{2} \right\} - 1 \rightarrow \text{wahl si } \hat{\alpha} + \hat{\beta} x > 0$$

$$= 2 \mathbb{1} \{ \hat{\alpha} + \hat{\beta} x > 0 \} - 1$$

$$\begin{array}{c} \hat{\alpha} + \hat{\beta} x = 0 \\ \textcircled{+} \quad \textcircled{-} \end{array}$$

• Analyse Discriminante Linéaire (Fisher, 1930).

$$\gamma(x) = \mathbb{P}(Y=+1|x).$$

Comment décrire  $(X, Y)$  ?

- Loi de  $X$  ( $F(\text{disc})$ ) +  $\gamma(x)$
- $p = \mathbb{P}(Y=+1)$ ,  $G = \mathbb{P}(X|Y=+1)$ ,  $H = \mathbb{P}(X|Y=-1)$ .

Par la loi de probabilités totale:  $F = pG + (1-p)H$

$$\begin{aligned} \mathbb{P}\{X \in A\} &= \mathbb{P}\{X \in A, Y=+1\} + \mathbb{P}\{X \in A, Y=-1\} \\ &= p \mathbb{P}\{X \in A | Y=+1\} + (1-p) \mathbb{P}\{X \in A | Y=-1\} \end{aligned}$$

$$\gamma(x) = \mathbb{E}[1\{Y=+1\}|x]$$

$$\mathbb{E}[\gamma(x)] = \mathbb{E}[1\{Y=+1\}] = \mathbb{P}(Y=+1) = p.$$

$$\bar{\Phi}(x) = \frac{dG}{dH}(x) = \frac{\mathbb{P}(X=x | Y=+1)}{\mathbb{P}(X=x | Y=-1)} \quad \text{Rappel proba. conditionnelle: } \mathbb{P}(A|B) = \frac{\mathbb{P}(B|A)}{\mathbb{P}(B)}.$$

$$= \frac{\mathbb{P}(Y=+1 | X=x)}{\mathbb{P}(Y=-1 | X=x)} \cdot \frac{\frac{\mathbb{P}(X=x)}{\mathbb{P}(Y=+1)}}{\frac{\mathbb{P}(X=x)}{\mathbb{P}(Y=-1)}} = \frac{\gamma(x) / p}{(1-\gamma(x)) / (1-p)} = \frac{1-p}{p} \cdot \frac{\gamma(x)}{1-\gamma(x)}$$

$$\hookrightarrow \boxed{\frac{p \bar{\Phi}(x)}{(1-p) + p \bar{\Phi}(x)} = \gamma(x)}$$

si  $\bar{\Phi}(x)$  rapport de vraisemblance connu.

### Hypothèses gaussiennes sous-jacentes:

$$\left. \begin{array}{l} G = N_A(\mu_+, \Gamma_+) \\ H = N_B(\mu_-, \Gamma_-) \end{array} \right\} \text{Lois caractérisées par les deux premiers moments}$$

Vecteur gaussien:  $X \sim N(\mu, \sigma^2)$ ,  $N(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$ ,  $\mathbb{E}(X) = \mu$ ,  $\text{Var}(X) = \sigma^2$

Vecteur gaussien:  $X = (x_1 \dots x_d) \Rightarrow \forall u \in \mathbb{R}^d$ ,  $\langle u, X \rangle = \sum_{i=1}^d u_i x_i$  est une v.a. gaussienne.

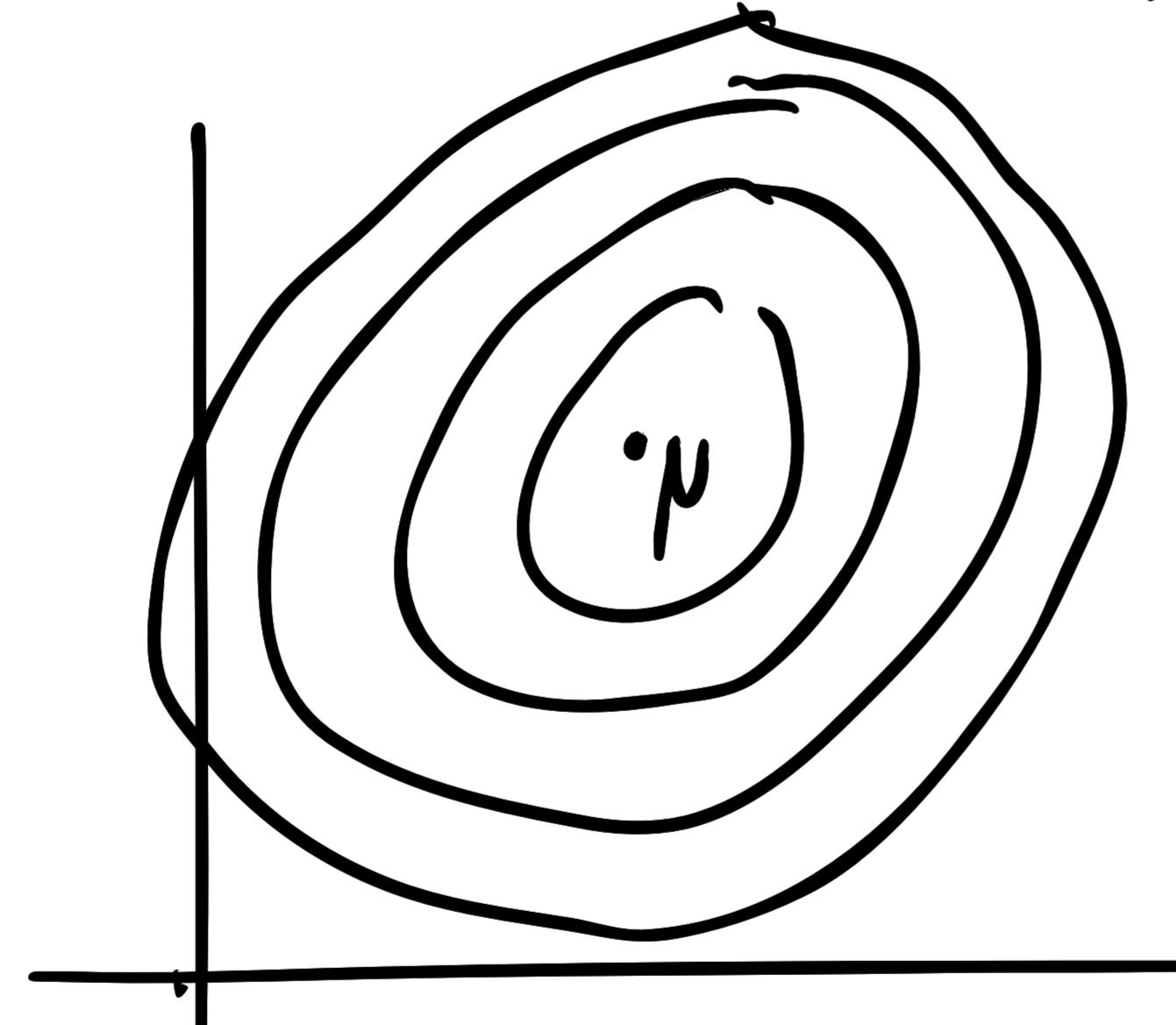
$$\hookrightarrow \langle X, \mu \rangle \sim N(\langle \mathbb{E}(X), \mu \rangle, {}^t \mu \Gamma \mu) \text{ où } \Gamma = \text{Var}(X) = \text{Cov}(X_i, X_j)$$

$\nearrow$   
 $\text{Var}(\langle X, \mu \rangle) \geq 0$

Si  $\Gamma$  est définie positive,  $\Gamma$  est inversible et le vecteur gaussien  $X$  a une densité sur  $\mathbb{R}^d$ .

$$x = (x_1 \dots x_d) \mapsto \frac{1}{(2\pi \det(\Gamma))^{d/2}} \exp\left(-\frac{1}{2} {}^t(x - \mathbb{E}(X)) \Gamma^{-1} (x - \mathbb{E}(X))\right)$$

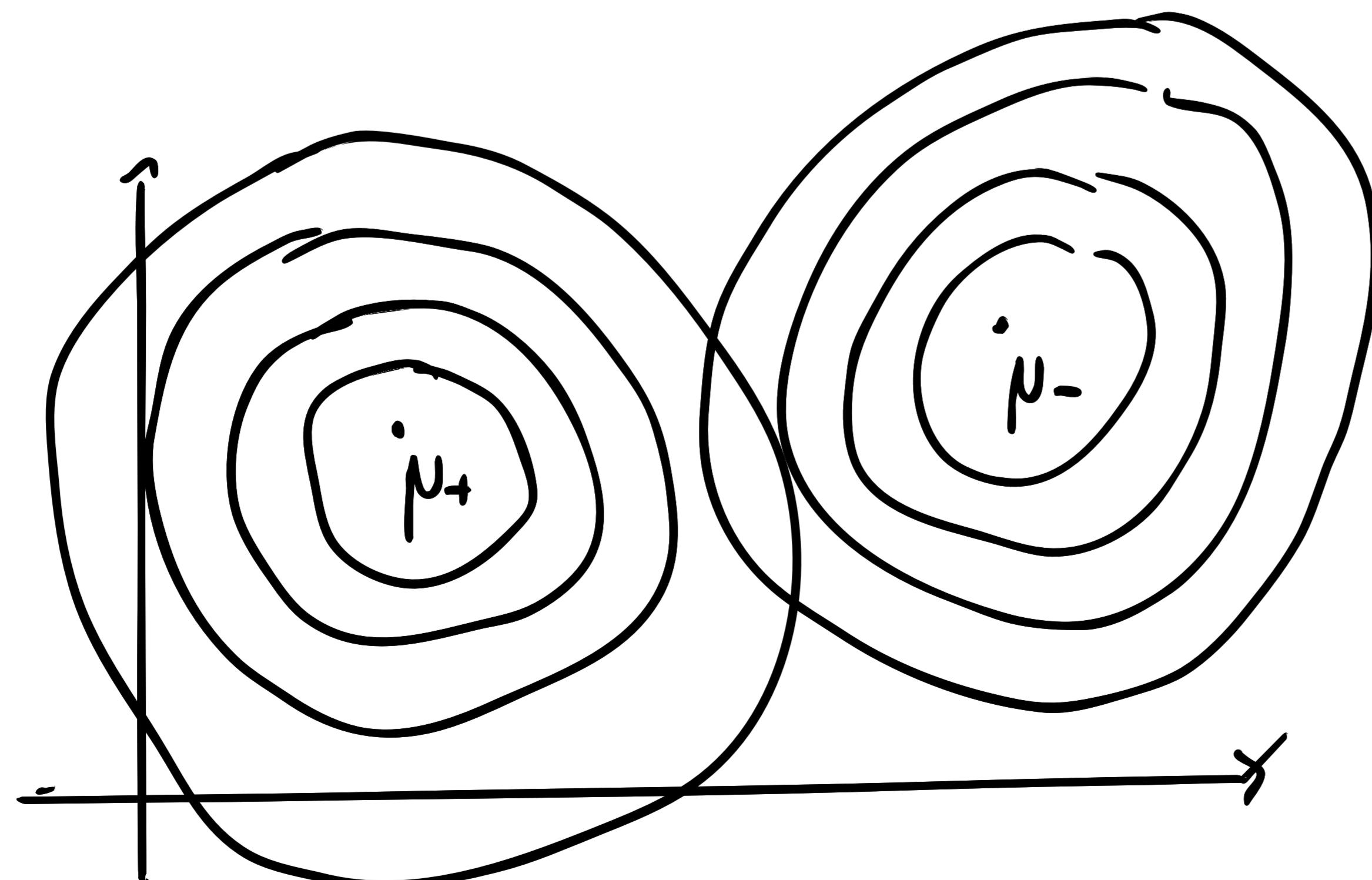
On suppose que les matrices de covariance avec la bale positive ( $\Gamma_+$ ) et négative ( $\Gamma_-$ ) sont similaires. En dimension 2, les gaussiennes ressemblent à des ellipses:



L'analyse en discriminant linéaire reste une approche plug-in:

$$\hat{\Phi}(x) = \frac{dG}{dT}(x) = \frac{1-p}{p} \cdot \frac{y(x)}{1-y(x)}$$

$$y(x) > \frac{1}{2} \Leftrightarrow \hat{\Phi}(x) > \frac{1-p}{p}$$



Si  $\Gamma_+ = \Gamma_- = \Gamma$ :

$$\underline{\Phi}(x) = \frac{\exp\left(-\frac{1}{2}^t(x-\mu_+)\Gamma^{-1}(x-\mu_+)\right)}{\exp\left(-\frac{1}{2}^t(x-\mu_-)\Gamma^{-1}(x-\mu_-)\right)}$$

$$= \exp\left(-\frac{1}{2}^t(x-\mu_+)\Gamma^{-1}(x-\mu_+) + \frac{1}{2}(x-\mu_-)\Gamma^{-1}(x-\mu_-)\right)$$

$$= \exp\left(^t x \Gamma^{-1} \mu_+ - \frac{1}{2}^t \mu_+ \Gamma^{-1} \mu_- - ^t x \Gamma^{-1} \mu_- + \frac{1}{2}^t \mu_- \Gamma^{-1} \mu_+\right).$$

$$\text{Si } \underline{\Phi}(x) > \frac{p}{1-p} \Leftrightarrow {}^t x \Gamma^{-1} (\mu_+ - \mu_-) + \frac{1}{2} (^t \mu_- \Gamma^{-1} \mu_- - {}^t \mu_+ \Gamma^{-1} \mu_+) \geq \log \frac{p}{1-p}$$

$$\Leftrightarrow \alpha + {}^t \beta x \geq 0$$

$$\text{où : } \beta = \Gamma^{-1}(\mu_+ - \mu_-)$$

$$\alpha = \frac{1}{2} (^t \mu_- \Gamma^{-1} \mu_- - {}^t \mu_+ \Gamma^{-1} \mu_+) - \log\left(\frac{p}{1-p}\right)$$

Alors comment trouver  $\alpha$  et  $\beta$ ? On les estime en trouvant  $\mu_+$ ,  $\mu_-$ ,  $\Gamma$  et  $p$ :  
 $(x_1, y_1), \dots, (x_n, y_n)$ , et  $n = n_+ + n_-$  nombre d'observations par label.

$$\bullet \hat{p} = \frac{\sum_{i=1}^n \mathbb{1}\{y_i = +1\}}{n} = \frac{n_+}{n}$$

$$\bullet \hat{\mu}_+ = \frac{1}{n_+} \sum_{\substack{i=1 \\ \{y_i=+1\}}}^n x_i \quad (\text{estime } \mu_+ = \mathbb{E}(X|Y=+1))$$

$$\bullet \hat{\mu}_- = \frac{1}{n_-} \sum_{\substack{i=1 \\ \{y_i=-1\}}}^n x_i$$

Rappel :  $Z_1, \dots, Z_n \stackrel{\text{iid}}{\sim} Z$ ,  $\mathbb{E}(Z^2) < +\infty$  (caractérisable)

$$\mathbb{E}(Z) \leftarrow \frac{1}{n} \sum_{i=1}^n Z_i = \bar{Z}_n$$

$$\text{Var}(Z) \leftarrow \frac{1}{n} \sum_{i=1}^n (Z_i - \bar{Z}_n)^2$$

$$\text{Cov}(Z, Z') = \mathbb{E}(ZZ') - \mathbb{E}(Z)\mathbb{E}(Z') = \mathbb{E}((Z - \mathbb{E}(Z))(Z' - \mathbb{E}(Z')))$$

$$\hookrightarrow \hat{\text{Cov}}(Z, Z') = \frac{1}{n} \sum_{i=1}^n Z_i Z'_i - \bar{Z}_n \bar{Z}'_n$$

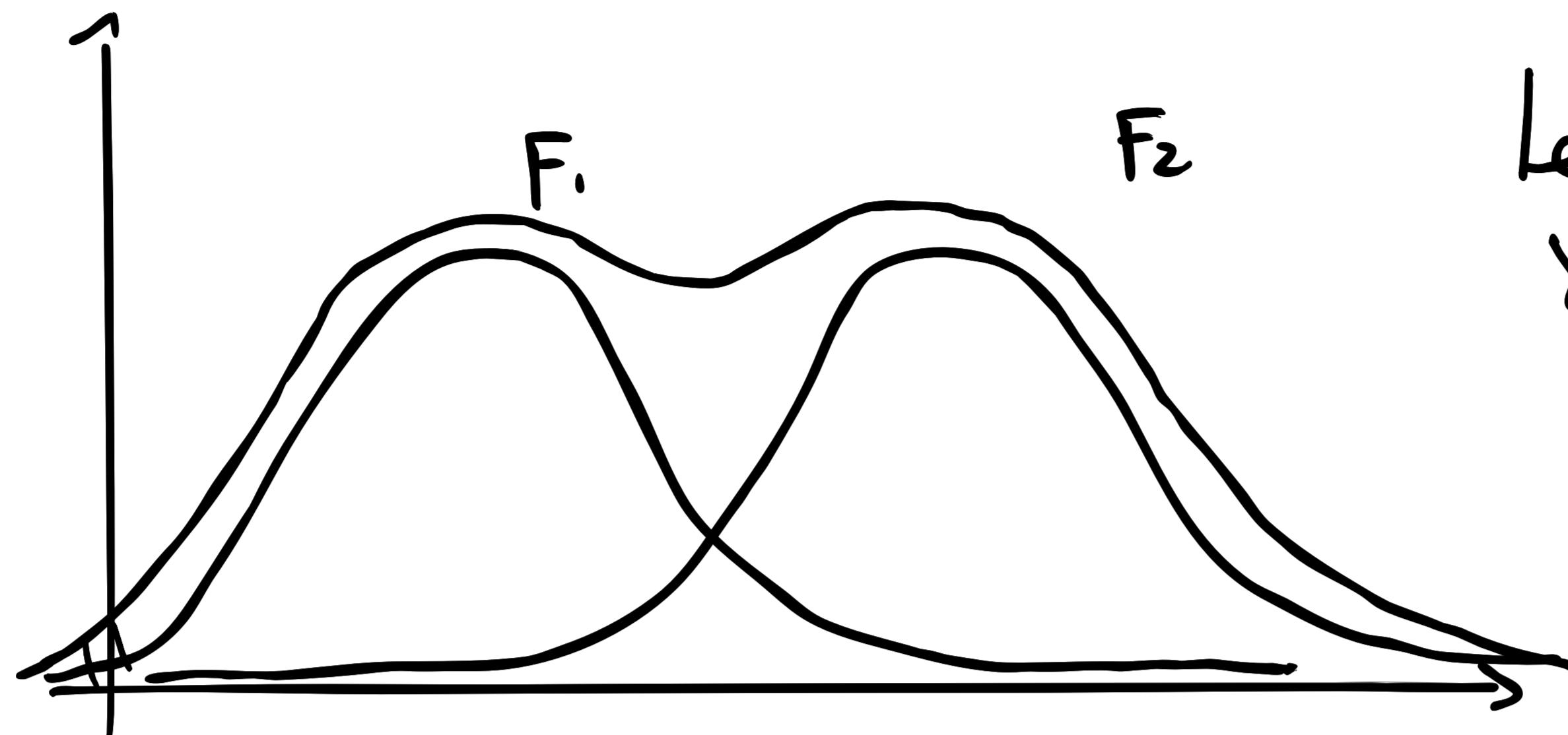
$$\bullet \hat{\Gamma} = \frac{n_+}{n} \hat{\Gamma}_+ + \frac{n_-}{n} \hat{\Gamma}_-$$

$\Rightarrow$  Nous permet de trouver  $\hat{\alpha}$  et  $\hat{\beta}$ .

- Quadratic discriminant analysis (QDA).

Relâcher l'hypothèse purement gaussienne et œuvrer aux mélanges de lois gaussiennes:

$$\delta F_1 + (1-\delta) F_2, \quad \delta \in (0, 1).$$



Le mélange devient une loi bi-modale.

$$\gamma = P(Y=1) = 1 - P(Y=2).$$

$$L(Z|Y=1) = F_1$$

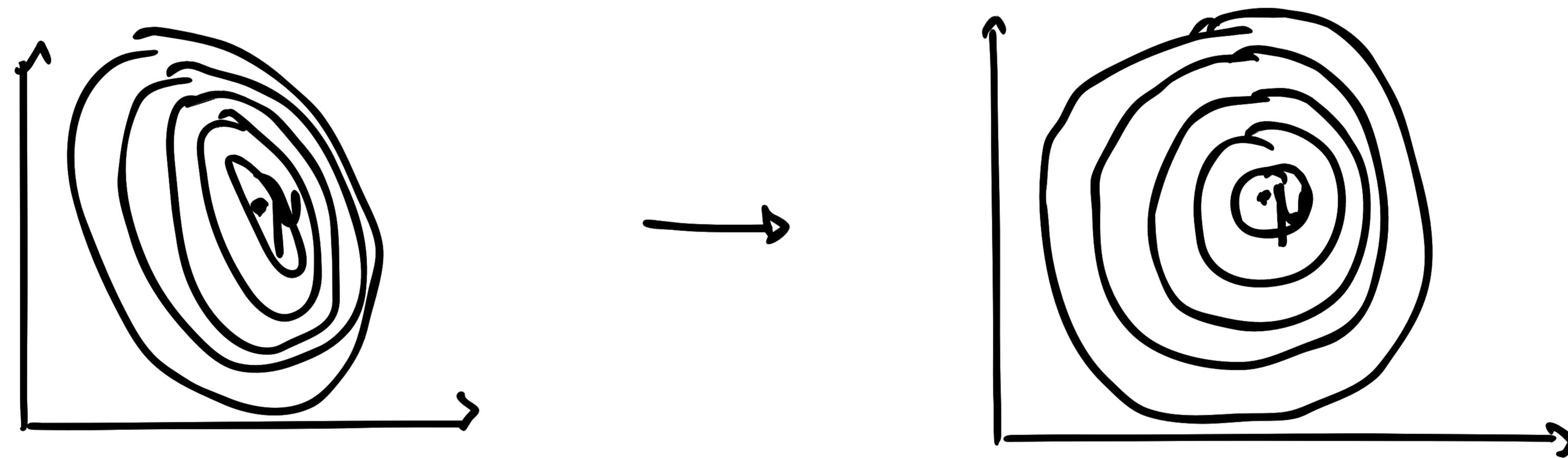
$$L(Z|Y=2) = F_2.$$

Ainsi, G et H peuvent désormais être bi-modales.

- Naive Bayes classification.

On suppose que  $\Gamma$  est diagonale:  $\Gamma = \text{diag } \sigma_i^2$ .

Approche relativement simpliste:



Les ellipses deviennent des cercles.

- Plus proches voisins (K-nearest neighbors).

$$x \in \mathbb{R}^d$$

D métrique / distance / dissimilarité.

↳ Des observations proches au sens de D devraient avoir la même étiquette.

D est une distance: axiomes

$$\begin{cases} D: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R} \\ D(x, x') = D(x', x) \\ D(x, x') \leq D(x, x'') + D(x'', x') \\ D(x, x') = 0 \iff x = x' \end{cases}$$

Exemple: métrique euclidienne:  $x = (x^{(1)}, \dots, x^{(d)})$

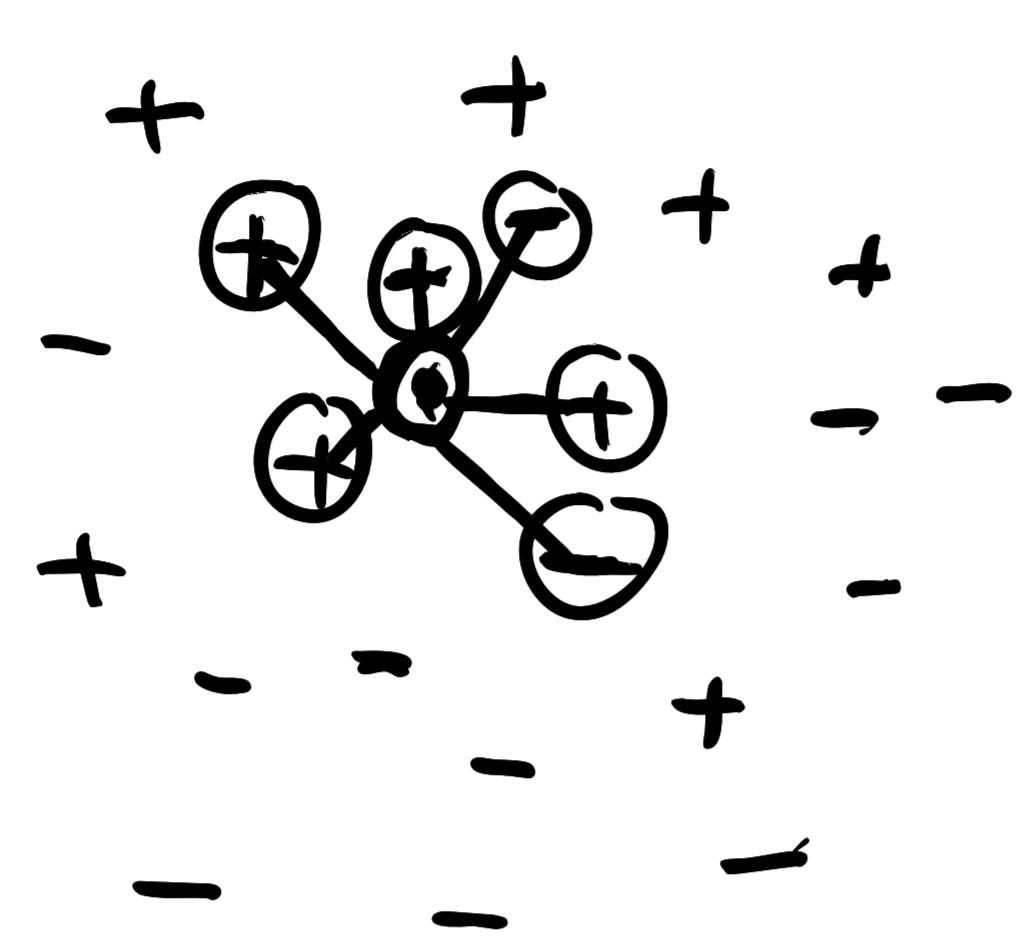
$$D(x, x') = \sqrt{(x^{(1)} - x'^{(1)})^2 + \dots + (x^{(d)} - x'^{(d)})^2}$$

↳ Sensible à l'unité de mesure. Nécessite une normalisation initiale.

Lecture conseillée : Encyclopédie des distances, David Boga.

Si  $\Gamma$  symétrique définie positive :

$$D_\Gamma(x, x') = \sqrt{+(x-x')^\top \Gamma^{-1}(x-x')}$$



$$g_{KNN}(x) = ?$$

$\sigma_x \in S_n$  groupe symétrique d'ordre  $n$ . opérat. permutation  
↳ N'est pas commutatif :  $\Sigma \neq \sigma + \varepsilon$

Le rôle de  $\sigma_x$  est de classer les points autour de  $x$  par distance :

$$D(x, X_{\sigma_x(1)}) \leq D(x, X_{\sigma_x(2)}) \leq \dots \leq D(x, X_{\sigma_x(n)})$$

Ne garde que les  $K$  plus proches voisins.

↳  $g_{KNN}(x) = g_{KNN}(x, Y_{\sigma_x(1)}, \dots, Y_{\sigma_x(k)})$  Si classificateurs de cette forme,  $g_{KNN}(x)$  minimise le risque empirique  $\hat{L}_n(g)$   
Comment classifier  $x$  ?

$$\begin{cases} +1 & \text{si } \sum_{i=1}^k [Y_{\sigma_x(i)}] > \frac{k}{2} \\ -1 & \text{si } \sum_{i=1}^k [Y_{\sigma_x(i)}] < k/2 \end{cases} \rightarrow \begin{array}{l} \text{Si } k \text{ est pair, il peut y avoir} \\ \text{égalité, Tirage au sort dans ce cas là.} \end{array}$$

Problèmes : • extrêmement sensible à la métrique  $D$   
• au choix de  $K$

Deux cas extrêmes : •  $k=1 \rightarrow$  beaucoup trop localisé  
•  $k=N \rightarrow$  trop peu localisé.

Limites : • il y a toujours qqch de plus proche, mais si peu de densité, ce n'est pas judicieux d'aller chercher très loin les voisins  
• coûteux en calcul car il faut garder en mémoire toutes les distances  
• ne passe pas à l'échelle

L'algorithme CART permet d'uniformiser la densité sur des grilles dans le cadre KNN.  
On peut utiliser les KNN pour le cadre de Régression (en prenant le centre de gravité des  $k$ -voisins les plus proches  $\Rightarrow$  la moyenne).

Si au lieu de prendre une perte  $L_2$  du type:  $\mathbb{E}(Z) = \arg\min \mathbb{E}(Z - c)^2 \rightarrow$  moyenne  
on prend une perte  $L_1$ :  $\arg\min \mathbb{E}[|Z - c|]$ , on prendrait la médiane.

Pour résumer, la meilleure façon de minimiser le risque empirique  $\hat{L}_n(g) = \frac{1}{k} \sum_{i=1}^k \mathbb{1}\{Y_i \neq g(x_i)\}$ .  
c'est le vote majoritaire.

Le choix de  $k$  est un problème de sélection de modèle

$$\min_{g \in \mathcal{G}_h} \hat{L}_n(g) \Rightarrow \hat{g}_{n,k}$$

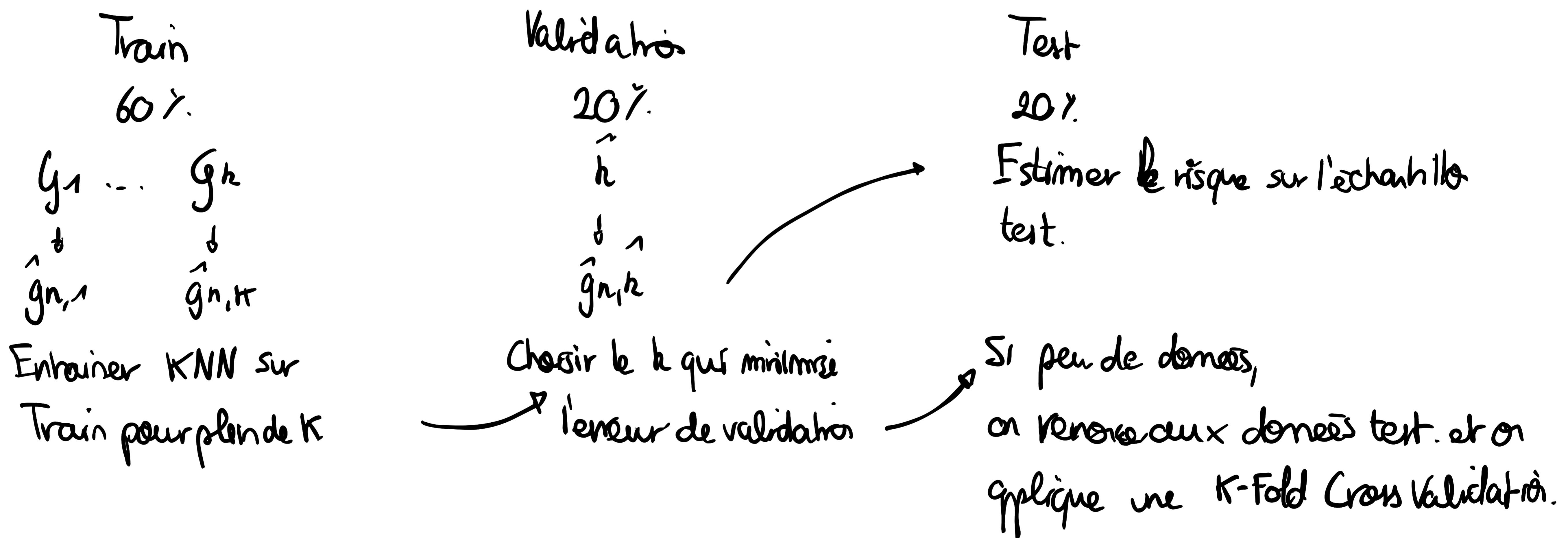
Attention:  $\min \hat{L}_n(\hat{g}_{n,k})$  est un mauvais critère (renvoie  $k=1$ )

$$\text{car } \mathbb{E}[\hat{L}_n(\hat{g}_{n,k})] \neq \mathbb{E}[L(\hat{g}_n)]$$

$$\text{fonction des données } x_i, y_i : \hat{L}_n(\hat{g}_{n,k}) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{Y_i \neq \hat{g}_{n,k}(x_i)\}$$

En pratique, comment choisir  $k$  ?

- Big Data  $\rightarrow$  suffisamment de données
- Ré-échantillonnage (Validation croisée, bootstrap).



## Theorie de la reconnaissance de forme

$(x, y)$ ,  $x$  entrée  $\in \mathcal{X}$   
 $y$  sortie  $\in \{-1, +1\}$ .  
 $L(g) = P(Y = g(x))$   
 $g: \mathcal{X} \rightarrow \{-1, +1\}$ .

où le problème de minimisation est: on minimise  $L(g) = g^*(x) = \begin{cases} +1 & \text{si } y(x) \geq \frac{1}{2} \\ -1 & \text{sinon} \end{cases}$

$L^* = L(g^*) \leq L(g) \Rightarrow 0 \leq L(g) - L^*$  est l'excès de risque de  $g$  et doit être contrôlé.

$$\begin{aligned} L(g) &= \mathbb{E}_x [y(x) \cdot \mathbf{1}_{\{g(x) = +1\}} + (1 - y(x)) \cdot \mathbf{1}_{\{g(x) = -1\}}] \\ L^* &= \mathbb{E} [\min \{y(x), 1 - y(x)\}] \leq \frac{1}{2} \\ &\leq \min \left( \underbrace{\mathbb{E}[y(x)]}_{p}, \underbrace{1 - \mathbb{E}[y(x)]}_{1-p} \right) \leq \min(p, 1-p). \end{aligned}$$

$$\text{où } p = P(Y = 1) = 1 - P(Y = -1)$$

Que vaut l'excès de risque ?

$$\begin{aligned} L(g) - L^* &= \mathbb{E} [y(x) (\mathbf{1}_{\{g(x) = -1\}} - \mathbf{1}_{\{g^*(x) = -1\}}) \\ &\quad + (1 - y(x)) (\mathbf{1}_{\{g(x) = +1\}} - \mathbf{1}_{\{g^*(x) = +1\}})] \\ &= \mathbb{E} [(2y(x) - 1) (\mathbf{1}_{\{g(x) = -1\}} - \mathbf{1}_{\{g^*(x) = -1\}})] \geq 0. \\ &= \mathbb{E} [2y(x) - 1 \mid \mathbf{1}_{\{g(x) \neq g^*(x)\}}] \end{aligned}$$

$$\Rightarrow L(g) - L^* = \mathbb{E} [2y(x) - 1 \mid \mathbf{1}_{\{g(x) \neq g^*(x)\}}].$$

Par la méthode du plug-in  $\hat{y}(x)$ ,  $g = 2 \mathbf{1}\{\hat{y}(x) \geq \frac{1}{2}\} - 1$

$$\Rightarrow L(g) - L^* = 2 \mathbb{E} [1_{\{y(x) \leq \frac{1}{2}\}} \mathbf{1}_{\{g(x) \neq g^*(x)\}}]$$

Pour  $x$  fixé, supposons  $g^*(x) = +1$  et  $g(x) = -1$ , c'est à dire:  
 $y(x) \geq \frac{1}{2}$  et  $\hat{y}(x) \leq \frac{1}{2}$ .

$$\hookrightarrow y(x) - \frac{1}{2} \leq y(x) - \hat{y}(x).$$

$$\begin{array}{c} \xleftarrow{\hspace{1cm}} \\ \hat{y}(x) \quad \frac{1}{2} \quad y(x) \\ \xrightleftharpoons{\hspace{1cm}} \end{array} \Rightarrow |y(x) - \frac{1}{2}| \mathbb{1}\{g(x) \neq g^*(x)\} \leq |y(x) - \hat{y}(x)|$$

Par le plug-in:  $L(g) - L^* = 2\mathbb{E}[|y(x) - \frac{1}{2}| \mathbb{1}\{g(x) \neq g^*(x)\}] \leq 2\mathbb{E}[|\hat{y}(x) - y(x)|]$   
 Nous offre une borne universelle qui ne dépend pas de la loi de  $y$ .

Les cas les plus problématiques sont les cas où  $y(x)$  est proche de  $1/2$ .

#### • Minimisation du risque empirique: (ERM)

$$(X_1, Y_1), \dots, (X_n, Y_n) \stackrel{iid}{\sim} (X, Y)$$

Fondé sur le principe fréquentiste, on doit observer  $n$  suffisamment grand.

Idee: Appliquer la minimisation du risque empirique sur une seule classe et observer l'effet sur l'excès de risque.

Nous disposons de ces données d'apprentissage, et la classe  $\mathcal{G}$  de classifier (i.e choix de l'algorithme et des hyper-paramètres).

ERM: On remplace  $L(g)$  inconnu par le risque empirique.

$$\hat{L}_n(g) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{Y_i \neq g(x_i)\} = L_{\hat{P}_n}(g) \quad \text{avec } \hat{P}_n = \frac{1}{n} \sum_{i=1}^n \delta_{(x_i, y_i)} \text{ masse de Dirac}$$

$$\hookrightarrow L_p(g) = \mathbb{E}_{(x, y) \sim P} [\mathbb{1}\{Y \neq g(x)\}]$$

donne proba  $1/n$  à toutes les observations.

NB: Soit  $a \in E$  masse de Dirac en  $a$ .

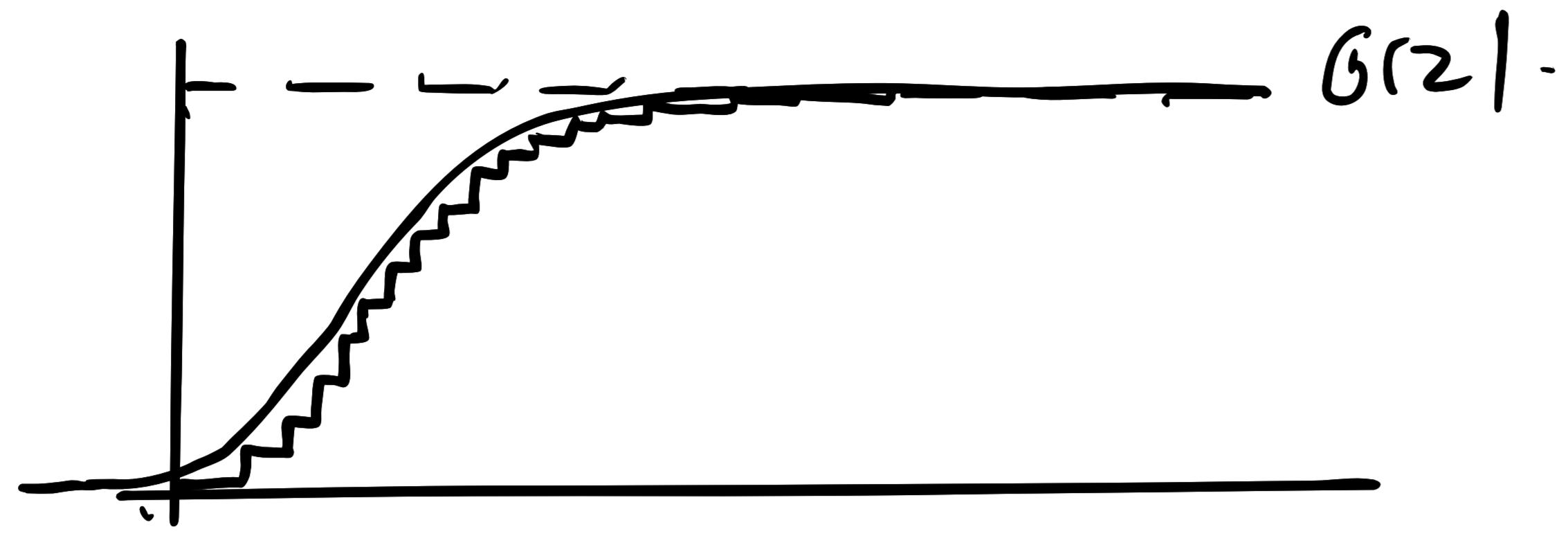
$$\text{Soit } \delta_a : F \subset E \mapsto \delta_a(F) = \begin{cases} 1 & \text{si } a \in F \\ 0 & \text{sinon} \end{cases}$$

V.  $a$  réelle  $Z \sim Q \Rightarrow \mathbb{E}_Q(Z)$  moyenne théorique

$$\hookrightarrow \text{moyenne empirique: } \bar{Z}_n = \frac{1}{n} \sum_{i=1}^n Z_i = \mathbb{E}_{Q_n}(Z), \quad Q_n = \frac{1}{n} \sum_{i=1}^n \delta_{Z_i}$$

$$\hookrightarrow \text{fonction de répartition: } \beta \mapsto P(Z < \beta) = G(z) = \mathbb{E}_Q[\mathbb{1}\{Z < \beta\}], \quad \beta \in \mathbb{R}$$

$$\text{fonction empirique: } \gamma \mapsto \hat{G}(z) = \mathbb{E}_{Q_n}(\mathbb{1}\{Z < \beta\}) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{Z_i < \beta\}$$



On veut réduire la distance entre les deux fonctions (Kolmogorov-Smirnov).

De retour à notre problème:  $\min_{g \in \mathcal{G}} \hat{L}_n(g) \Rightarrow \hat{g}_n$ .

Il peut y avoir plusieurs minimiseurs du risque empirique  $\hat{g}_n$ . On en considère un seul.  $\hat{g}_n(x) = \hat{g}_n(x_1, (x_1, y_1), \dots, (x_n, y_n))$  considéré comme aléatoire car il dépend de:  $(x_1, y_1), \dots, (x_n, y_n)$ .

Qu'est-ce que mesure  $\hat{g}_n(x)$ ? C'est le produit de l'expérience aléatoire  $(x_1, y_1), \dots, (x_n, y_n) \stackrel{iid}{\sim} (X, Y)$  de distribution  $P$ .

$$\Rightarrow L(\hat{g}_n) = P_{(x,y)} \{Y \neq \hat{g}_n(x)\}$$

Soit  $(X, Y)$  nouvelle donnée indépendante de  $(x_1, y_1), \dots, (x_n, y_n)$ :

$$L(\hat{g}_n) = P(Y \neq \hat{g}_n(X) \mid (x_1, y_1), \dots, (x_n, y_n)).$$

Le on souhaite obtenir un excès de risque petit en ayant appris sur nos données d'entraînement.

L'excès de risque du minimiseur du risque empirique :  $L(\hat{g}_n) - L^*$

Soit  $\bar{g} \in \arg \min_{g \in \mathcal{G}} \tilde{L}(g)$  le minimiseur du risque théorique de la classe  $\mathcal{G}$ .

Si  $\bar{g}$  contenait le minimiseur de Bayes,  $\bar{g} = g^*$ . Mais ce n'est généralement pas le cas.

$$\begin{aligned} \Rightarrow L(\hat{g}_n) - L^* &= L(\hat{g}_n) - \hat{L}_n(\hat{g}_n) \rightarrow \text{inconnu} \\ &+ \hat{L}_n(\hat{g}^*) - \hat{L}_n(\bar{g}) \Rightarrow 0 \text{ car } \hat{g}_n \in \arg \min_{g \in \mathcal{G}} \hat{L}_n(g) \text{ et } \bar{g} \in \mathcal{G} \Rightarrow \hat{L}_n(\hat{g}_n) \leq \hat{L}_n(\bar{g}) \\ &+ \hat{L}_n(\bar{g}) - L(\bar{g}) \rightarrow \text{inconnu} \\ &+ L(\bar{g}) - L^* \Rightarrow L(\bar{g}) - L^* = \inf_{g \in \mathcal{G}} L(g) - L^* = \text{Biais} \end{aligned}$$

$$L(\hat{g}_n) - L^* \leq 2 \sup_{g \in \mathcal{G}} |L_n(g) - L(g)| + \inf_{g \in \mathcal{G}} L(g) - L^*$$

Si  $\mathcal{G}$  trop simple,  $L$  ne sera pas dedans, le biais sera élevé mais le sup (l'apprentissage) sera très simple

Quand est-ce que l'apprentissage marche ?

↳ Quand  $\sup_{g \in \mathcal{G}} |\hat{L}_n(g) - L(g)| \Rightarrow$  doit être vrai uniformément sur la classe  $\mathcal{G}$ .

↳ Renvoie à la notion de processus empirique :  $\{\hat{L}_n(g) - L(g)\}_{g \in \mathcal{G}}$  processus empirique - une collection de moyennes empiriques.

↳  $\sup_{g \in \mathcal{G}} |\hat{L}_n(g) - L(g)| = \|V\|$  où  $V$  v.a. dans un espace de sortes / fonctions de  $\mathcal{G}$ .

Pour en savoir plus : "Probability in Banach spaces".

Penons un cas fini :  $\#\mathcal{G} = N < +\infty$ .

Soit un résultat du type :  $\forall \delta \in (0, 1)$  seuil de confiance.

↳ avec probabilité  $\geq 1 - \delta$ ,  $L(\hat{g}_n) - \inf_{g \in \mathcal{G}} L(g) \leq ?$

$$\leq 2 \max_{g \in \mathcal{G}} |\hat{L}_n(g) - L(g)|$$

borne supérieure de confiance.

$$P(A \cup B) \leq P(A) + P(B)$$

$$t > 0 : P\{\max_{g \in \mathcal{G}} |\hat{L}_n(g) - L(g)| > t\}$$

$$= P\left(\bigcup_{g \in \mathcal{G}} \{\hat{L}_n(g) - L(g) > t\}\right) \leq \sum_{g \in \mathcal{G}} P\{\hat{L}_n(g) - L(g) > t\}$$

↑ inégalité de déviation

Quelle est l'inégalité de déviation la plus simple ?

$Z$  v.a. réelle  $\geq 0$ ,  $E[Z] > 0$

• Inégalité de Markov :  $P(Z > t) \leq \frac{E(Z)}{t}$  (Inégalité de déviation)

$$Z > t \mathbb{1}(Z > t)$$

$$E(Z) \geq t P(Z > t)$$

• Inégalité de Chebyshev :  $p \geq 1$ ,  $E[|Z|^p] < +\infty$

$$P(|Z - E(Z)| \geq t) = P(|Z - E(Z)|^p \geq t^p) \leq \frac{E(|Z - E(Z)|^p)}{t^p}$$

• Inégalité exponentielle de W. Hoeffding.

Les inégalités de déviation sont démontrées par la méthode de Chernoff

$Z_1, \dots, Z_n \stackrel{iid}{\sim} Z$  v.a. à valeurs réelles,  $E(Z) < +\infty$ ,  $E(Z) = 0 \Rightarrow$  Cas imp. av. Mkr.

$$P\left(\sum_{i=1}^n Z_i \geq t\right) = P\left(e^{s \sum_{i=1}^n Z_i} - e^{st}\right) \leq e^{-st + \log E[e^{s \sum_{i=1}^n Z_i}]}$$

↑ Markov

$$\Rightarrow P\{\sum_{i=1}^n Z_i \geq t\} \leq \exp\left(\inf_{s>0} \{-st + \log E[e^{s \sum_{i=1}^n Z_i}]\}\right)$$

$$\text{Or, } \log \mathbb{E}[e^{s\sum_i Z_i}] = n \log \mathbb{E}[e^{sZ}]$$

- Rappel : • Transformation de Fourier :  $\mathbb{E}[e^{sZ}]$   
 • Transformation de Laplace :  $\mathbb{E}[e^{sz}]$ .

Cela revient à contrôler la log bplace.

$$P(\sum_i Z_i > t) \leq \exp\left(\inf_{s>0} (-st + \log \mathbb{E}[e^{sz}])\right)$$

Lemme : Soit  $Z$  v.a réelle de paixance 0. ,  $a \leq Z \leq b$ .

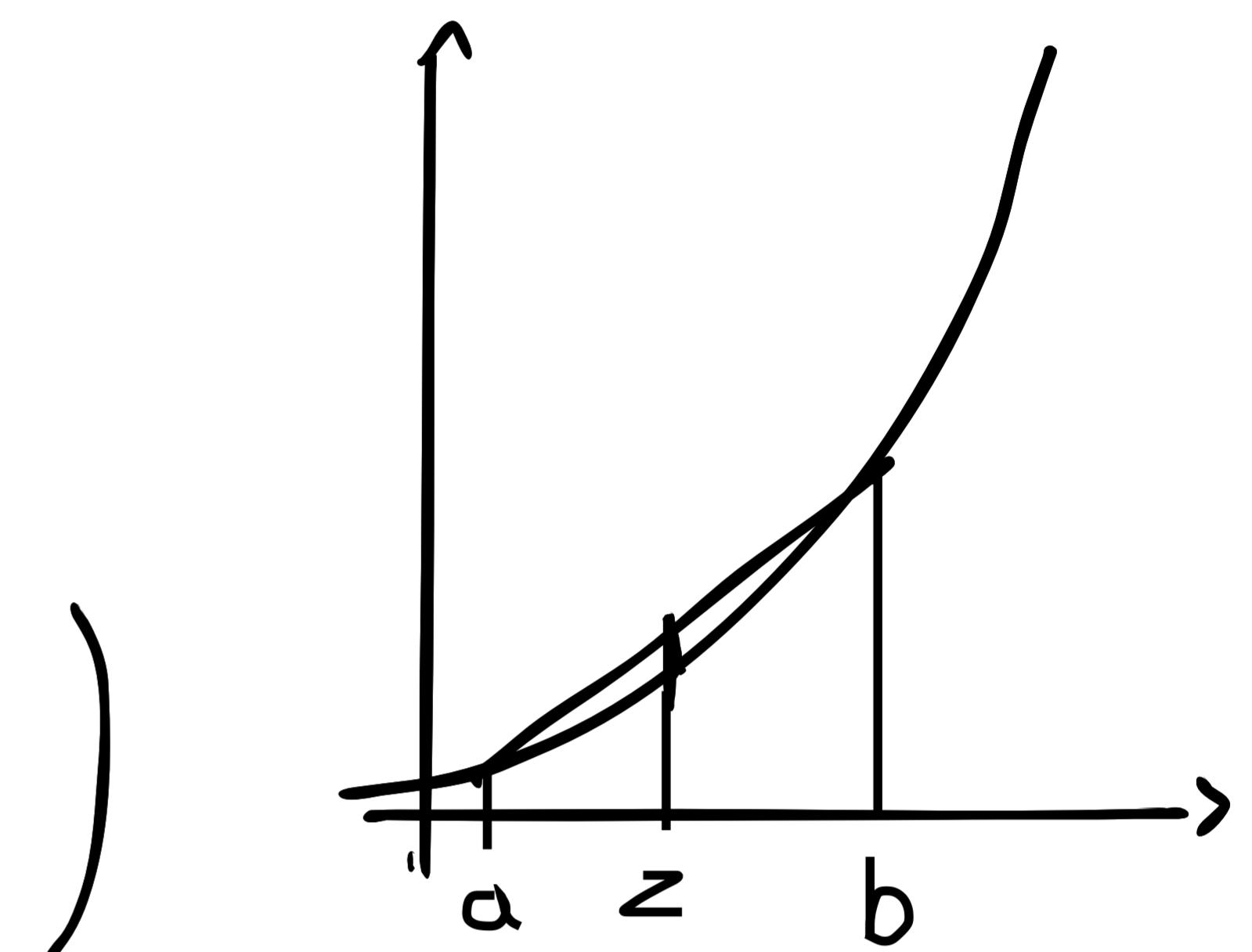
$$\text{Alors } \forall s > 0, \mathbb{E}[e^{sz}] \leq e^{sz} (b-a)^2/8$$

Resssemble à une leplace gaussienne.

Preuve :  $s > 0$ ,  $u \mapsto e^u$  convexe.

$$\forall z \in (a, b), \left( \frac{z-a}{b-a} e^{sa} \leq e^{sz} \leq e^{sb} \frac{b-z}{b-a} \right)$$

$$e^{sz} \leq e^{sa} \frac{b-z}{b-a} + e^{sb} \frac{z-a}{b-a}$$



$$\text{Or, } a \leq z \leq b \text{ p.s., } \mathbb{E}(Z)=0 \Rightarrow \mathbb{E}(e^{sz}) \leq \frac{be^{sa} - ae^{sb}}{b-a}$$

$$\text{Soit } p = \frac{-a}{b-a} \Rightarrow \mathbb{E}(e^{sz}) = e^{-ps(b-a)} \left( 1 - p + pe^{s(b-a)} \right) = e^{\Phi(u)}$$

Pour  $\Phi(u) = -pu + \log(1 - p + pe^u)$  → Développement de Taylor.

$$\Phi(0) = 0$$

$$\Phi'(u) = -p + \frac{pe^u}{1-p+pe^u} \rightarrow \Phi'(0) = 0$$

$$\Phi''(u) = \frac{p(1-p)}{(1-p+pe^u)^2} = \left( \frac{p(1-p)e^{-u}}{(1-p)e^{-u} + p} \right) \leq \frac{1}{4}$$

$$\text{Car } \frac{CD}{(C+D)^2} \leq \frac{1}{4} \Rightarrow 4CD \leq (CD)^2 \Rightarrow 0 \leq (C-D)$$

$$\text{Taylor d'ordre 2: } \Phi(u) = \Phi(0) + u\Phi'(0) + \frac{u^2}{2}\Phi''(0)$$

$$\Rightarrow \mathbb{E}(e^{sz}) = e^{\Phi(0)} \leq e^{u^2/8} \quad \checkmark$$

Comment appliquer ceci au log log place ?

$$\begin{aligned} \forall s > 0, \quad a_i \leq z_i \leq b_i \text{ p.s., } \mathbb{E}(Z) = 0 \\ \mathbb{P}\left(\sum_{i=1}^n Z_i \geq t\right) &\leq \exp(-st + n \log e^{s^2(b-a)^2/8}) \\ &= \underbrace{\exp(-st + ns^2(b-a)^2/8)}_{\text{minimum pour } -t + \frac{ns}{4}(b-a)^2 = 0} \\ &\hookrightarrow s = 4t/(n(b-a)^2) \end{aligned}$$

Théorème : Inégalité de Hoeffding

$$\left( \text{Si } a_i \leq z_i \leq b_i \Rightarrow s = \frac{4t}{\sum_{i=1}^n (b_i - a_i)^2} \right)$$

Sont  $Z_1, \dots, Z_n$  indépendants entre.

$$\forall i \in \{1, \dots, n\}, \quad a_i \leq z_i \leq b_i \text{ p.s.}$$

$$\text{Alors } \forall t > 0, \quad \mathbb{P}\left(\sum_{i=1}^n Z_i \geq t\right) \leq \exp\left(\frac{-2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right)$$

$$\Rightarrow \mathbb{P}\left(\sum_{i=1}^n Z_i \leq -t\right) \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right)$$

$\hookrightarrow \mathbb{P}\left(\sum_{i=1}^n -Z_i \geq t\right) \Rightarrow -b_i \leq Z_i \leq a_i \Rightarrow$  la borne  $(b_i - a_i)^2$  ne change pas.

On parle d'inégalité de déviation sous-gauzinne.

Si on remplace  $t^2$  par  $t$ , c'est une ineq. de dev. sous-poissonne.

$$\Rightarrow \mathbb{P}\left\{|\sum_{i=1}^n Z_i| \geq t\right\} \leq 2 \exp\left(\frac{-2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right) \quad (*).$$

Retour à l'analyse du principe de minimisation du risque empirique (ERM) ( $g=N$ ).

$$\mathbb{P}\left(\sup_{g \in \mathcal{G}} |\hat{L}_n(g) - L(g)| \geq t\right) \leq \sum_{g \in \mathcal{G}} \mathbb{P}\{|\hat{L}_n(g) - L(g)| \geq t\}$$

$$\hat{L}_n(g) - L(g) = \frac{1}{n} \sum_{i=1}^n \underbrace{\{1_{\{Y_i \neq g(X_i)\}} - L(g)\}}_{Z_i, \quad a=-1, b=+1}$$

$$\text{D'après } (*): \quad \mathbb{P}\{|\hat{L}_n(g) - L(g)| \geq t\} \leq 2 \exp\left(-\frac{1}{2}nt^2\right)$$

$\hookrightarrow$  le risque empirique devient proche du risque théorique pour autant que  $n$  soit grand.

$$\Rightarrow \sum_{g \in \mathcal{G}} \mathbb{P}\{|\hat{L}_n(g) - L(g)| \geq t\} \leq 2N e^{-\frac{1}{2}nt^2}$$

$$\Rightarrow \frac{nt^2}{2} \leq \log\left(\frac{2N}{\delta}\right), \text{ où } \delta = 2N e^{-\frac{1}{2}nt^2}$$

$$t = \sqrt{\frac{2 \log(2N/\delta)}{n}}.$$

Avec probabilité plus grande que  $1-\delta$ :  $\sup_{g \in \mathcal{G}} |\hat{L}_n(g) - L(g)| = \sqrt{\frac{2 \log(2N/\delta)}{n}}$   
et donc:  $L(\hat{g}_n) - L^* \leq 2\sqrt{\frac{2 \log(2N/\delta)}{n}} + \inf_{g \in \mathcal{G}} L(g) - L^*$

↳ C'est un résultat sous forme de borne supérieure de cofrince.

Parfois, on souhaite un résultat sous forme de moyenne:

Soit notre but de contrôler:  $\mathbb{E}[L(\hat{g}_n) - L^*]$

$$\leq 2\mathbb{E}[\sup_{g \in \mathcal{G}} |\hat{L}_n(g) - L(g)|] + \inf_{g \in \mathcal{G}} L(g) - L^*$$

Rappel:  $\mathbb{E}(Z) = \int_0^\infty P(Z > t) dt$  (IPP)

$$\leq 2N \int_{t=0}^{e^{\frac{1}{2}nt^2}} e^{\frac{1}{2}nt^2} dt + \inf_{g \in \mathcal{G}} L(g) - L^*$$

↓

$$= \frac{1}{2} \frac{\sqrt{2\pi(1/n^2)}}{\sqrt{2\pi(1/n^2)}} \int_{t=-\infty}^{+\infty} e^{-\frac{1}{2\sqrt{\frac{1}{n^2}}}t^2} dt \quad \begin{matrix} \text{1 car guérison} \\ \text{pour } \sigma = \frac{1}{\sqrt{n}} \end{matrix}$$

(dimo par coordonnées polaires  $I^2=1$ )

$$\leq N \frac{\sqrt{2\pi}}{\sqrt{n}} + \inf_{g \in \mathcal{G}} L(g) - L^*$$

Lemma: Soient  $Z_1, \dots, Z_n$  des v.a. centrées.

①  $\forall s > 0, \mathbb{E}(e^{sZ_i}) \leq e^{\sigma^2 s^2/2}$  avec  $\sigma > 0$ .  $\forall i \in \{1, \dots, n\}$

$$\text{Alors } \mathbb{E}[\max_{1 \leq i \leq n} |Z_i|] \leq \sigma \sqrt{2 \log N}$$

② Si de plus,  $\forall s > 0, \forall i \in \{1, \dots, N\}$ :

$$\mathbb{E}(e^{-sZ_i}) \leq e^{\sigma^2 s^2/2}$$

$$\text{Alors } \mathbb{E}(e^{-sZ_i}) \leq e^{\sigma^2 s^2/2} \Rightarrow \mathbb{E}[\max_{1 \leq i \leq n} |Z_i|] \leq \sigma \sqrt{2 \log(2N)}$$

Faut-il démontrer 1 et 2?

$\mathbb{E}[\max |Z_i|] = \mathbb{E}[\max \{-Z_1, \dots, -Z_n, Z_1, \dots, Z_n\}] \Rightarrow$  ① entraîne ②. Démontrer ② seulement.

Démonstration de ②:

$$\forall s > 0. \text{ Par l'inégalité de Jensen: } e^{s\mathbb{E}[\max_{1 \leq i \leq N} Z_i]} \leq \mathbb{E}[e^{s \max_i Z_i}] = \mathbb{E}[\max_i e^{s Z_i}]$$
$$\leq \mathbb{E}\left[\sum_{i=1}^N e^{s Z_i}\right] = \sum_N \mathbb{E}[e^{s Z_i}]$$
$$\leq N e^{s^2 \sigma^2 / 2}$$

On passe au log:  $\mathbb{E}(\max_i Z_i) = \frac{\log N}{s} + \frac{\sigma^2 s}{2}$  minimum pour  $s = \sqrt{\frac{2 \log N}{\sigma^2}}$ . ✓

Dans l'ERM:  $\mathbb{E}[e^{s(L(g) - L(\hat{g}))}] \leq e^{s^2/2n^2}$

D'après le lemme 1:  $\hat{L}(g) - L(g) = Z$

$$\mathbb{E}[L(\hat{g}) - L(g)] \leq 2 \sqrt{\frac{2 \log(2N)}{n}} + \inf_{g \in \mathcal{G}} L(g) - L^*$$

↑ toujours problématique si n petit, mais bien moins rapide car log.

Jusqu'ici, nous avons supposé que  $\# \mathcal{G} = N$  fini. Que se passe-t-il quand  $\mathcal{G}$  est infini ?

### Fondements théoriques.

1) Inégalité de McDiarmid : Différences bornées.

Repose sur l'inégalité de martingale : Azuma (1950)

2) Inégalité de Vapnik - Chervonenkis.

$$\mathbb{E} [\sup_{g \in \mathcal{G}} |\hat{L}_n(g) - L(g)|]$$

1) But: Contrôle de  $\mathbb{P} (\sup_{g \in \mathcal{G}} |\hat{L}_n(g) - L(g)| - \mathbb{E} [\sup_{g \in \mathcal{G}} |\hat{L}_n(g) - L(g)|] > t)$   
 $\forall t > 0$ .

Sans aucune hypothèse sur: - la loi de  $(X, Y)$   
- la classe de  $\mathcal{G}$ .

Notons que  $\sup_{g \in \mathcal{G}} |\hat{L}_n(g) - L(g)|$  est une quantité dépendant de  $(x_1, y_1), \dots, (x_n, y_n)$  de façon très complexe.

Théorème:

- Soit  $x_1, \dots, x_n$  indépendants, à valeurs dans  $E_1, \dots, E_n$  respectivement
- Soit  $f: \prod_{i=1}^n E_i \rightarrow \mathbb{R}$  vérifiant  $(x_1, \dots, x_n) \mapsto f(x_1, \dots, x_n)$  l'hypothèse des différences bornées
- $\forall i \in \{1, \dots, n\}, c_i < +\infty$
- $\forall x_i' \in E_i:$

$$|f(x_1, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_n) - f(x_1, \dots, x_{i-1}, x_i', x_{i+1}, \dots, x_n)| \leq c_i$$

Alors  $\forall t > 0$ ,

- $\mathbb{P}[|f(x_1, \dots, x_n) - \mathbb{E}[f(x_1, \dots, x_n)]| \geq t] \leq e^{-2t^2 / \sum_{i=1}^n c_i^2}$
- $\mathbb{P}[|f(x_1, \dots, x_n) - \mathbb{E}[f(x_1, \dots, x_n)]| \leq -t] \leq e^{-2t^2 / \sum_{i=1}^n c_i^2}$
- $\mathbb{P}[|f(x_1, \dots, x_n) - \mathbb{E}[f(x_1, \dots, x_n)]| > t] \leq 2e^{-2t^2 / \sum_{i=1}^n c_i^2}$

Ce sont des inégalités sous-gaussiennes.

Preuve: Soient  $V$  et  $Z$  deux v.a.,  $\mathbb{E}[V|Z] = 0$  p.s.

et  $h(Z) \leq V \leq h(Z) + C$ , p.s où  $h$  est une fonction et  $C > 0$  une constante.  
 $\mathbb{E}(e^{sV}|Z) \leq e^{s^2 C^2 / 8}$ .

Pour rappel, Preuve de Hoeffding :  $\mathbb{E}[V] = 0$ ,  $a \leq V \leq b$ ,  $\mathbb{E}[e^{sV}] \leq e^{s^2(b-a)^2/8}$ .

Notre problème n'est rien d'autre qu'un problème de Hoeffding conditionnel.

forme de somme telescopique.

On pose :  $V = f(x_1 \dots x_n) - \mathbb{E}[f(x_1 \dots x_n)] = \sum_{i=1}^n V_i$   
 avec  $\forall i \in \{1 \dots n\}$ ,  $V_i = \underbrace{\mathbb{E}[f(x_1 \dots x_n) | x_1 \dots x_i]}_{H_i(x_1 \dots x_i)} - \underbrace{\mathbb{E}[f(x_1 \dots x_n) | x_1 \dots x_{i-1}]}_{\int H_i(x_1 \dots x_{i-1}, x_i) F_i dx_i}$   
 où  $F_i$  désigne la loi de  $X_i$ .

Soyons  $w_1$  et  $w_2$  des v.a.,  $\mathbb{E}[K(w_1, w_2)^2] < +\infty$ .

Si  $w_1, w_2$  indépendants :  $\mathbb{E}[K(w_1, w_2) | w_2] = k(w_2)$  avec  $\forall w$  :

$$k(w) = \mathbb{E}_{w_1}[K(w_1, w)]$$

Soyons  $W_i = \sup_{\mu} \{ H_i(x_1 \dots x_{i-1}, \mu) - \int H_i(x_1 \dots x_{i-1}, x_i) F_i dx_i \}$

$Z_i = \inf_{\mu} \{ H_i(x_1 \dots x_{i-1}, \mu) - \int H_i(x_1 \dots x_{i-1}, x_i) F_i dx_i \}$ .

$Z_i \leq \gamma_i \leq W_i$  p.s et  $W_i - Z_i \leq C_i$  p.s par différences bornées.

D'après le lemme,  $\forall s > 0$ ,  $\mathbb{E}(e^{sV_i} | X_1 \dots X_{i-1}) \leq e^{s^2 C_i^2 / 8}$ .

$P(\sum_{i=1}^n V_i \geq t) \leq \exp(-st + \log \mathbb{E}(e^{s \sum_{i=1}^n V_i})) \rightarrow$  Chernoff  $\forall s > 0$ .

$$\begin{aligned} \mathbb{E}[e^{s \sum_{i=1}^n V_i}] &= \mathbb{E}\left[\underbrace{\mathbb{E}[e^{s \sum_{i=1}^n V_i} | X_1 \dots X_{n-1}]}_{e^{s \sum_{i=1}^n V_i} = e^{s \sum_{i=1}^{n-1} V_i} e^{s V_n}}\right] \\ &= \mathbb{E}\left[e^{s \sum_{i=1}^{n-1} V_i} \underbrace{\mathbb{E}[e^{s V_n} | X_1 \dots X_{n-1}]}_{\leq e^{s^2 C_n^2 / 8} \cdot \mathbb{E}[e^{s \sum_{i=1}^n V_i}]}\right]. \end{aligned}$$

$$\leq e^{s^2 C_n^2 / 8} \times e^{s^2 C_{n-1}^2 / 8} \times \dots = e^{s^2 / 8 \sum_{i=1}^n C_i^2}$$

$\Rightarrow -st + \frac{s^2}{8} \sum_{i=1}^n C_i^2$ . Le minimum pour  $t = \frac{s}{4} \sum C_i^2$ .

$$s = \frac{4t}{\sum_{i=1}^n C_i^2}$$

On applique le résultat à :  $\sup_{g \in \mathcal{G}} |\hat{L}_n(g) - L(g)| = f((x_1, y_1), \dots, (x_n, y_n))$

qui est à differences bornées avec  $c_i = \frac{1}{n}$  (car fonction indicatrice, max 1 changement).

$$\forall t > 0, \mathbb{P} \left\{ \sup_{g \in \mathcal{G}} |\hat{L}_n(g) - L(g)| - \mathbb{E} \left[ \sup_{g \in \mathcal{G}} |\hat{L}_n(g) - L(g)| \right] \geq t \right\} \leq e^{-2t^2/n \cdot (\frac{1}{n})^2} = e^{-2nt^2} \underbrace{\delta}_{\delta}$$

Ainsi, avec probabilité  $1-\delta$ :

$$\sup_{g \in \mathcal{G}} |\hat{L}_n(g) - L(g)| \leq \mathbb{E} \left[ \sup_{g \in \mathcal{G}} |\hat{L}_n(g) - L(g)| \right] + \sqrt{\frac{\log(1/\delta)}{2n}}$$

Et donc : si  $\hat{g}_n \in \operatorname{argmin}_{g \in \mathcal{G}} \hat{L}_n(g)$ :

$$L(\hat{g}_n) - L^+ \leq 2 \mathbb{E} \left[ \sup_{g \in \mathcal{G}} |\hat{L}_n(g) - L(g)| \right] + \sqrt{2 \log(1/\delta)/n} + \inf_{g \in \mathcal{G}} L(g) - L^+$$

Reste à contrôler

2) Contrôle de :  $\mathbb{E} \left[ \sup_{g \in \mathcal{G}} |\hat{L}_n(g) - L(g)| \right] \leq ?$

$$\hat{L}_n(g) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}\{y_i \neq g(x_i)\}.$$

$$g \rightarrow A_g = \{x : g(x) = +1\}.$$

$$g(x) = 2 \mathbf{1}\{x \in A_g\} - 1$$

$$\hat{L}_n(g) = \frac{1}{n} \sum_{i=1}^n \left[ \mathbf{1}\{(x_i, y_i) \in A_g \times \{-1\}\} + \mathbf{1}\{(x_i, y_i) \in A_g^c \times \{+1\}\} \right]$$

Inégalité de Vapnik - Chervonenkis :

Soient  $x_1, \dots, x_n$  à valeurs dans  $\mathbb{R}^d$ , iid de la loi (inconnue)  $\mu(\text{doc})$ :  $x_i \sim \mu(\text{doc})$

La distribution empirique est  $\hat{\mu}_n = \frac{1}{n} \sum_{i=1}^n \delta_{x_i}$ .

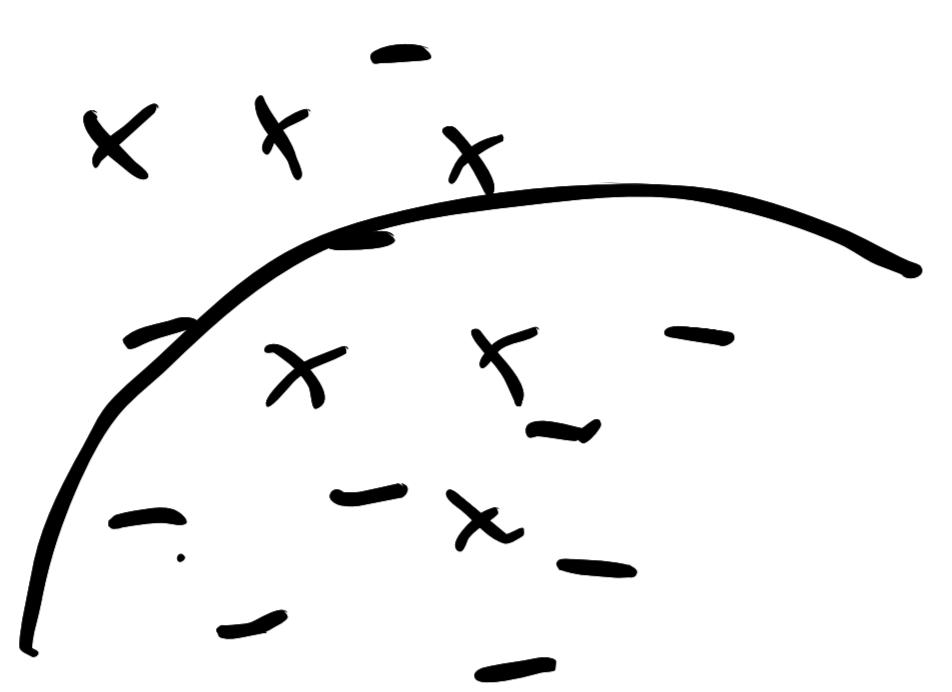
$\forall A \subset \mathbb{R}^d$  mesurable:  $\hat{\mu}_n(A) = \frac{1}{n} \sum_{i=1}^n \delta_{x_i}(A) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}\{x_i \in A\} \xrightarrow[n \rightarrow \infty]{\text{LFGN}} \mu(A)$  p.s

$\mathbb{E} \left[ \sup_{A \in \mathcal{A}} |\hat{\mu}_n(A) - \mu(A)| \right] \leq ?$ ,  $\mathcal{A}$  classe de sous ensembles mesurables de  $\mathbb{R}^d$ .

Il faut d'abord introduire le coefficient d'éclatement de la classe  $A$ :

Soyons  $x_1, \dots, x_n$  dans  $\mathbb{R}^d$ .

Trace de  $t$  sur  $\{x_1, \dots, x_n\}$ ,  $\text{Tr}_t(A) = \{A \cap \{x_1, \dots, x_n\} : A \in t\}$



$\#\text{Tr}_t(A) \leq 2^n$ . On dit que  $A$  éclate les  $n$  points  $x_1, \dots, x_n$  si  $\#\text{Tr}_t(A) = 2^n$ .

Le coefficient d'éclatement de  $t$  à l'ordre  $n$  est:  $S_t(n) = \sup_{\{x_1, \dots, x_n\}} \#\text{Tr}_t(A)$

La dimension de Vapnik - Chervonenkis (VC) de  $t$ :

$$\dim_{VC}(t) = \sup \{n > 1 : 2^n = S_t(n)\} \in \mathbb{N}^* \cup \{+\infty\}$$

On dit que  $A$  est de dimension de VC finie si:  $\dim_{VC}(A) < +\infty$

$A = \{[-\infty, x], x \in \mathbb{R}\} \cup \{[x, +\infty], x \in \mathbb{R}\}$  Classe de sous ensembles de  $\mathbb{R}$ .

2 points:



On ne peut jamais éclater ces 3 points.

3 points:



$$\Rightarrow \dim_{VC} A = 2$$

Dans  $\mathbb{R}^d$ ,  $A = \{\mu \in \mathbb{R}^d : d + {}^t \beta \mu > 0\} : d \in \mathbb{R}, \beta \in \mathbb{R}^d\}$ ,  $\dim_{VC} A = d + 1$ .

$\Rightarrow$  Inégalité de Vapnik - Chervonenkis:  $\mathbb{E}[\sup_{A \in t} |\hat{\mu}_n(A) - \mu(A)|] \leq c \sqrt{\frac{\dim_{VC} A}{n}}$

Preuve: 3 étapes:

1) Symétrie

2) Randomisation

3) Comptage.

On introduit un échantillon fantôme  $x'_1, \dots, x'_n \stackrel{iid}{\sim} \mathcal{N}$ , indép de  $x_1, \dots, x_n$ .

$$\mu(A) = \mathbb{E}[\hat{\mu}_n(A)] = \text{avec } \hat{\mu}_n(A) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}\{x_i \in A\}.$$

$$\begin{aligned} \mathbb{E}_{x_1, \dots, x_n} [\sup_{A \in t} |\hat{\mu}_n(A) - \mu(A)|] &= \mathbb{E}_{x_1, \dots, x_n} [\sup_{A \in t} |\hat{\mu}_n(A) - \mathbb{E}[\hat{\mu}_n'(A)]|] \\ &= \mathbb{E}_{x_1, \dots, x_n} [\sup_{A \in t} |\mathbb{E}[\hat{\mu}_n(A) - \hat{\mu}_n'(A)]|] \end{aligned}$$

$$\leq \mathbb{E}_{x_1 \dots x_n} \left[ \sup_{A \in \mathcal{A}} \mathbb{E}_{x'_1 \dots x'_n} \left[ |\hat{\mu}_n(A) - \hat{\mu}'_n(A)| \right] \right] \quad (\text{Inégalité de Tscheb})$$

$$\leq \mathbb{E}_{x_1 \dots x_n} \left[ \mathbb{E}_{x'_1 \dots x'_n} \left[ \sup_{A \in \mathcal{A}} |\hat{\mu}_n(A) - \hat{\mu}'_n(A)| \right] \right]$$

$$= \mathbb{E}_{\substack{x_1 \dots x_n \\ x'_1 \dots x'_n}} \left[ \sup_{A \in \mathcal{A}} |\hat{\mu}_n(A) - \hat{\mu}'_n(A)| \right] = \mathbb{E}_{\substack{x_1 \dots x_n \\ x'_1 \dots x'_n}} \left[ \sup_{A \in \mathcal{A}} \left| \frac{1}{n} \sum_{i=1}^n \{1\{x_i \in A\} - 1\{x'_i \in A\}\} \right| \right]$$

On considère un chaos de Rademacher:  $\varepsilon_1 \dots \varepsilon_n$  iid  $P\{\varepsilon_i = +1\} = 1 - P\{\varepsilon_i = -1\} = 1/2$

$$= \mathbb{E}_{\substack{x_1 \dots x_n \\ x'_1 \dots x'_n \\ \varepsilon_1 \dots \varepsilon_n}} \left[ \sup_{A \in \mathcal{A}} \left| \frac{1}{n} \sum_{i=1}^n \varepsilon_i (1\{x_i \in A\} - 1\{x'_i \in A\}) \right| \right]$$

$$= \mathbb{E}_{\substack{x_1 \dots x_n \\ x'_1 \dots x'_n}} \left[ \mathbb{E}_{\varepsilon_1 \dots \varepsilon_n} \left[ \sup_{A \in \mathcal{A}} \left| \frac{1}{n} \sum_{i=1}^n \varepsilon_i (1\{x_i \in A\} - 1\{x'_i \in A\}) \right| \mid x_1 \dots x_n, x'_1 \dots x'_n \right] \right]$$

Fixons des points  $x_i$  et  $x'_i$ :  $\rightarrow$  fixe dans l'espérance conditionnelle.

$$\mathbb{E}_{\varepsilon_1 \dots \varepsilon_n} \left[ \sup_{A \in \mathcal{A}} \left| \frac{1}{n} \sum_{i=1}^n \varepsilon_i (1\{x_i \in A\} - 1\{x'_i \in A\}) \right| \right]$$

prend moins de valeurs que  $S_A(2n)$   $\Rightarrow \# \hat{A} \leq S_A(2n)$ .

$$\mathbb{E} \left[ \max_{1 \leq i \leq n} |Y_i| \right] \leq \sqrt{2 \log(2N)}$$

$$\text{ssi } \mathbb{E}[e^{SY_i}] \leq e^{S^2 \sigma^2 / 2}$$

$$\mathbb{E} \left[ e^{S \frac{1}{n} \sum \varepsilon_i (1\{x_i \in A\} - 1\{x'_i \in A\})} \right] \leq e^{S^2 / 2n}$$

c.f Lemme sur le contrôle de la log Laplace.

$$\mathbb{E}_{\varepsilon_1 \dots \varepsilon_n} \left[ \sup_{A \in \mathcal{A}} \left| \frac{1}{n} \sum_{i=1}^n \varepsilon_i (1\{x_i \in A\} - 1\{x'_i \in A\}) \right| \right]$$

$$\leq \sqrt{\frac{2 \log(2S_A(2n))}{n}} \leq \sqrt{\frac{2 \log 2 + 2 \log S_A(n)}{n}} \leq \sqrt{\frac{2 \log 2 + 2 \log(n+1) \times V}{n}}$$

$$S_A(2n) \leq S_A(n)^2$$

$$S_A(n) \leq (n+1)^V$$

où  $V = \dim_{\mathbb{R}} A$

Même si  $G$  infini, la vitesse de convergence reste proche de  $\frac{1}{\sqrt{n}}$ .

**Conclusion :** Le minimiseur du risque empirique fournit pour les données d'entraînement.  
Mais est-il valable pour un nouvel échantillon  $(X, Y)$  ?  
Qui pour autant que  $Y$  ne soit pas complexe.  
• que  $\#G$  soit fini  
• ou non.  
Pour autant que la dimension VC soit finie.

Grauantes statistiques pour

- Binary decision trees
- general partitioning techniques with hypercubes
- linear separators ...
- And model selection !

Méthodes for:

- non linear SVM
- boosting
- random forest.

Comment répartir ses échantillons Train, Test, Validation ?

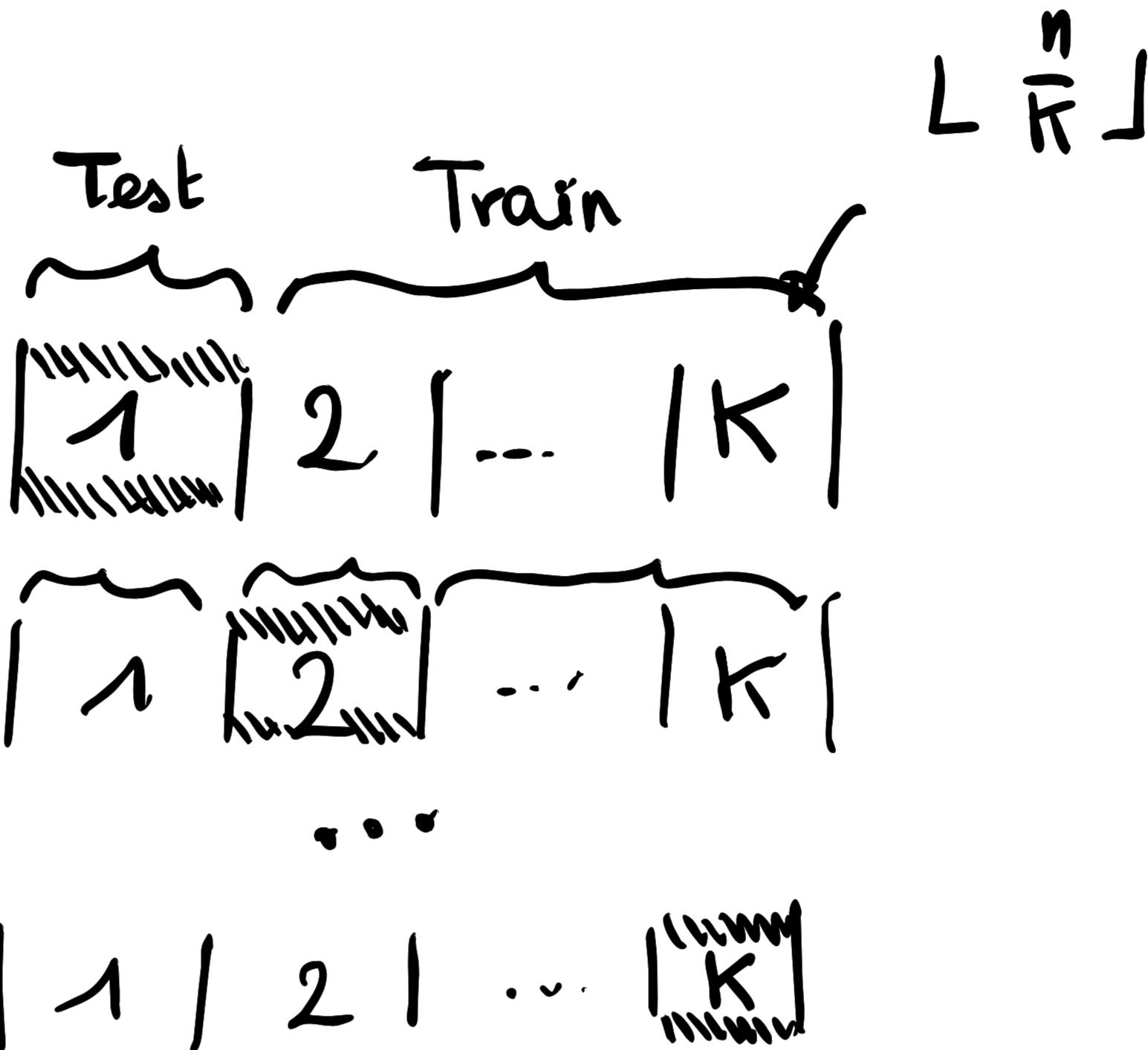
- Big Data

Train	Validation	Test
-------	------------	------

- Smart Data

K-fold Cross Validation

↳ Typically,  $K=5, 10\dots$



↳ If we take  $K=n$ , called Leave-One-Out.

On apporte des limitations sur la complexité de  $g$ :

$n=1, \dots, N$ , ERM sur  $\mathcal{G}_m$ .

$$\min_{g \in \mathcal{G}_m} \hat{L}_n(g) \Rightarrow \hat{g}(n)$$

↳  $\hat{L}_n(\hat{g}(n)) + \text{pen}(n, \mathcal{G}_m)$ .  $\Rightarrow$  Pénalisation de la complexité additive.

Pénalité idéale :  $\text{pen}^*(n, \mathcal{G}_m) = L(\hat{g}^{(n)}) - \hat{L}(\hat{g}^{(n)})$ .

$$\text{↳ } L(\hat{g}(n)) - \hat{L}_n(\hat{g}^{(n)}) \leq \sup_{g \in \mathcal{G}_m} |L_n(g) - L(g)| \leq C \sqrt{\frac{V_m}{n}}$$

Résumé : • Si  $\# g$  fini :  $\mathbb{E} [L(\hat{g}_n) - L(g)] \leq 2 \sqrt{\frac{2 \log(2N)}{n}} + \inf_{g \in \mathcal{G}} L(g) - L^*$

• Si  $\# g$  infini :  $\mathbb{E}_{\epsilon_1 \dots \epsilon_n} [\sup_{g \in \mathcal{G}} \left| \frac{1}{n} \sum_{i=1}^n \epsilon_i (1\{x_i \in A\} - 1\{x'_i \in A\}) \right|]$   
 $\leq \sqrt{\frac{2 \log 2 + 2 \log(n+1) V}{n}}$

↳ Vitesse de convergence proche de  $\frac{1}{\sqrt{n}}$ . On peut généraliser l'échantillon Train.

## Local Averaging - Decision Trees and regression trees.

$X$  vec. aléatoire dans  $\mathcal{X} = \mathbb{R}^p$   
 $Y$  target variable dans  $\mathcal{Y}$   $\Rightarrow Y = \begin{cases} \{1, -1\} & \text{classification binaire supervisée} \\ \{1, \dots, c\} & \text{multi-classe} \end{cases}$   
 $D$  joint probability  $(X, Y)$ .

$H$  la classe hypothétique : le jeu de modèles que l'on considère. (Garant)

Soit une fonction de perte  $l : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}^+$

On souhaite résoudre :  $\underset{f \in H}{\operatorname{argmin}} \mathbb{E}_{(x, y) \sim D}[l(Y, f(x))]$

d'après les données :  $\{(X_i, Y_i), i=1, \dots, n\}$

$\Leftrightarrow$  Minimiser le vrai risque sur la base d'un échantillon qui doit être représentatif.

### Fonctions cibles.

- \* Classification : Pour  $l(y, f(x)) : (0/1)$ , meilleur classifieur si les vraies pertes sont connues :  $f_{\text{Bayes}} = \arg \max_c P(Y=c | X)$
- \* Régression : Pour  $l(y, f(x)) = (y - f(x))^2$  perte L2, la meilleure solution est :  $f_{\text{reg}}(x) = \mathbb{E}(Y | X)$

Cependant, les probabilités réelles ne sont pas connues.

### Minimiser le risque empirique:

$S = \{(X_i, Y_i), i=1 \dots n\}$  iid et  $\Omega$  terme de régularisation.

$$\underset{f \in H}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^n l(Y_i, f(X_i)) + \Omega(f)$$

$\downarrow$   
contrôle la complexité du modèle.

### Apprentissage : Estimation fonctionnelle.

$f_n = t(S_n, H, l, \Omega)$  avec :

- $t$  : apprentissage / estimation / optimisation  $\rightarrow$  algorithme (contraintes ressources, temps)
- $S_n$  : données d'entraînement
- $H$  : classe de fonctions  $\rightarrow$  Plus les données sont préparées, plus l'algorithme est simple.
- $\Omega$  : mesure de complexité
- $l$  : fonction de perte locale  $\rightarrow$  Coût de chaque classe peut différer (pharma par ex.)

↳ Prédiction : Etant donné un nouveau "x", calculer  $f_n(x)$ .

## I. Rappel : K-plus-proches voisins (KNN)

Cas avec deux classes:

$$f_{KNN}(x) = \arg \max_{y \in \{-1, 1\}} \frac{N_y^K(x)}{K} \quad \text{avec:}$$

\*  $K$  entier  $> 0$

\*  $d$  métrique sur  $X \times X$  (distance euclidienne par exemple)

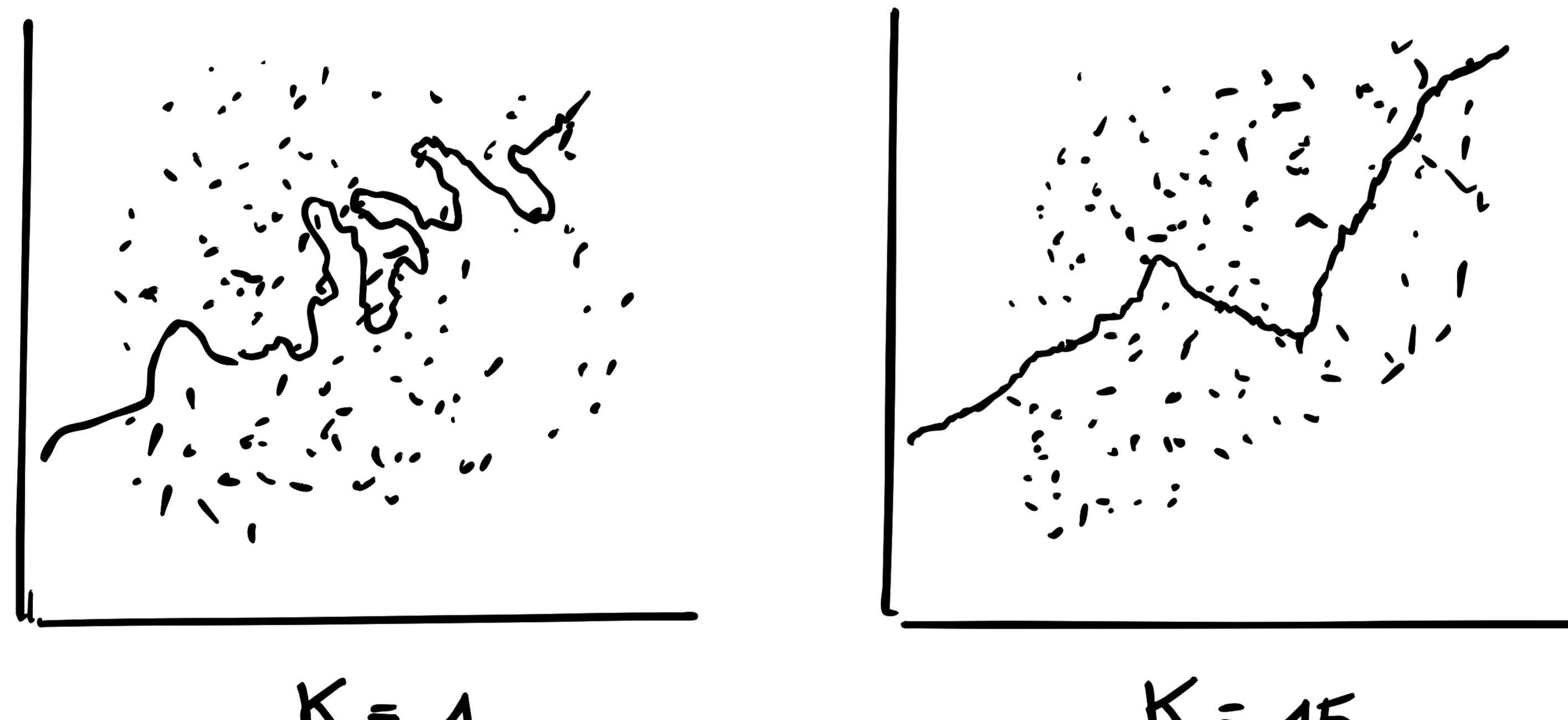
\*  $S = \{(x_i, y_i), i=1 \dots n\}$

\* Pour une donnée  $x$ , permutation d'indices dans  $\{1, \dots, n\}$  t.q:

$$d(x, x_{\sigma(1)}) \leq d(x, x_{\sigma(2)}) \leq \dots \leq d(x, x_{\sigma(n)})$$

\*  $S_x^K = \{x_{\sigma(1)}, \dots, x_{\sigma(K)}\}$  :  $K$  premiers voisins de  $x$ .

$$N_y^K(x) = |\{x_i \in S_x^K, y_i = y\}|$$



$K$  trop petit  $\rightarrow$  trop sensible  
 $K$  trop grand  $\rightarrow$  pas assez sensible.

Erreur de généralisation:

$$Y = f(x) + \varepsilon, \varepsilon \text{ centré de variance } \sigma_\varepsilon^2.$$

$$\mathbb{E}[(Y - \hat{f}(x))^2] = \mathbb{E}[Y^2 + \hat{f}(x)^2 - 2Y\hat{f}(x)]$$

$$= \mathbb{E}[Y^2] + \mathbb{E}[\hat{f}(x)^2] - 2\mathbb{E}[Y\hat{f}(x)]$$

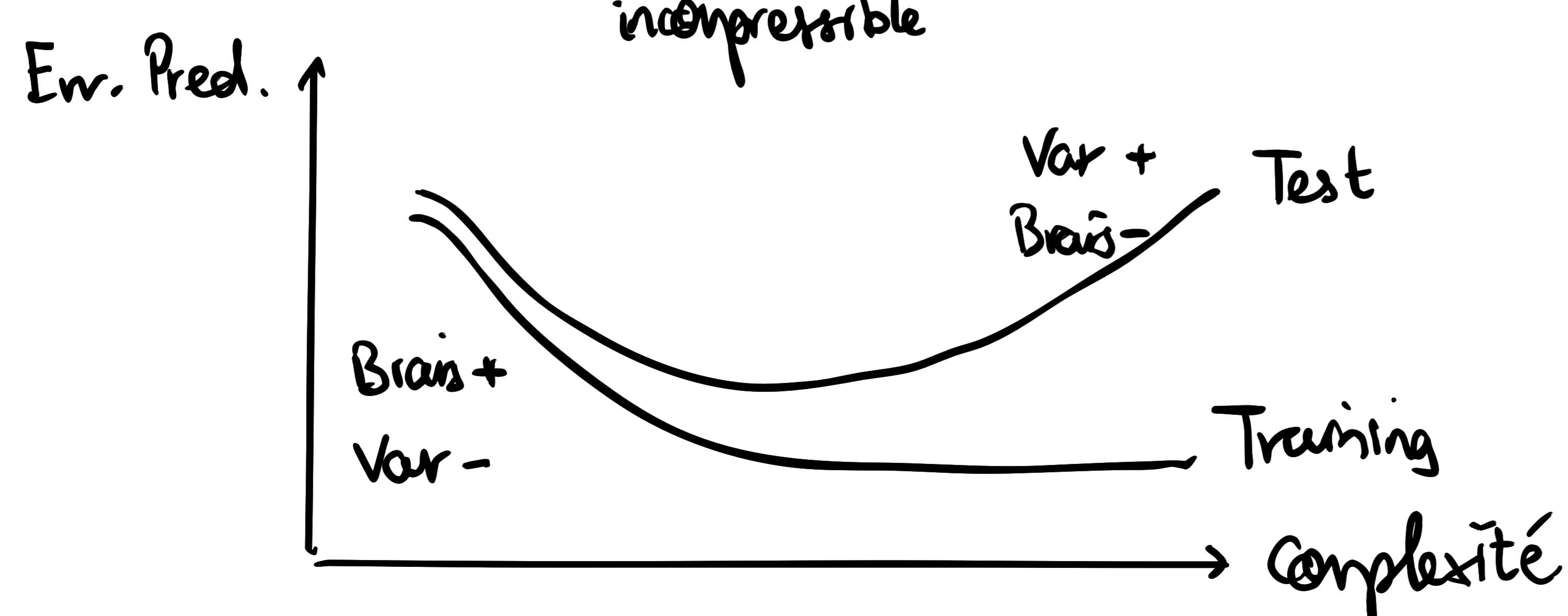
$$= \text{Var } Y + \mathbb{E}[Y]^2 + \text{Var } \hat{f}(x) + \mathbb{E}[\hat{f}(x)]^2 - 2\mathbb{E}[f(x) + \varepsilon]\mathbb{E}[\hat{f}(x)]$$

$$= \sigma_\varepsilon^2 + \mathbb{E}[f(x) + \varepsilon]^2 + \mathbb{E}[\hat{f}(x)]^2 - 2\mathbb{E}[f(x)]\mathbb{E}[\hat{f}(x)] + \text{Var } \hat{f}(x)$$

$$= \sigma_\varepsilon^2 + \mathbb{E}[f(x) - \hat{f}(x)]^2 + \text{Var } \hat{f}(x)$$

$$= \sigma_\varepsilon^2 + \underset{\uparrow}{\text{Biais}}^2 + \text{variance}$$

incompressible



Dans le cas KNN, la décomposition biais-variance s'écrit ainsi :

Posons  $x_0$ . Supposons que l'aléa ne viennent que des  $\epsilon$ . On peut montrer que :

$$\mathbb{E}[(y - \hat{f}(x_0))^2] = \sigma_\epsilon^2 + (f(x_0) - \frac{1}{K} \sum_{i=1}^K f(x_{i,0}))^2 + \frac{\sigma_\epsilon^2}{K} \quad \text{dans le cas régression}$$

↳ Si  $K \nearrow$ , Var  $\downarrow$ , mais biais  $\uparrow$

Dans les KNN, le choix de la mesure de distance est très important.

L'apprentissage ne coûte rien, mais le test très coûteux en temps de calcul.

## II. Modèle en moyenne locale

- Pour une classification binaire

Limite des KNN : un voisin peut être très loin de  $x$ . (flau de la dimension)

Coordonner une partition de l'espace des features :  $C_1 \cup \dots \cup C_K = \mathcal{X}$ .

On applique la règle de majorité : supposer que  $x$  est dans  $C_k$ .

1) Compter le nombre d'exemple d'entraînements avec un label positif dans  $C_k$ .

2) Si  $\sum_{i: x_i \in C_k} \mathbb{1}\{Y_i = +1\} > \sum_{i: x_i \in C_k} \mathbb{1}\{Y_i = -1\}$

Prédire  $Y = +1$ .

Autrement, prédire  $Y = -1$

→ si  $\begin{cases} > 0.5 \Rightarrow 1 \\ < 0.5 \Rightarrow 0 \end{cases}$

Cela correspond au classifieur "plug-in"  $2 \cdot \mathbb{1}\{\hat{n}(x)\} - 1$ , où :

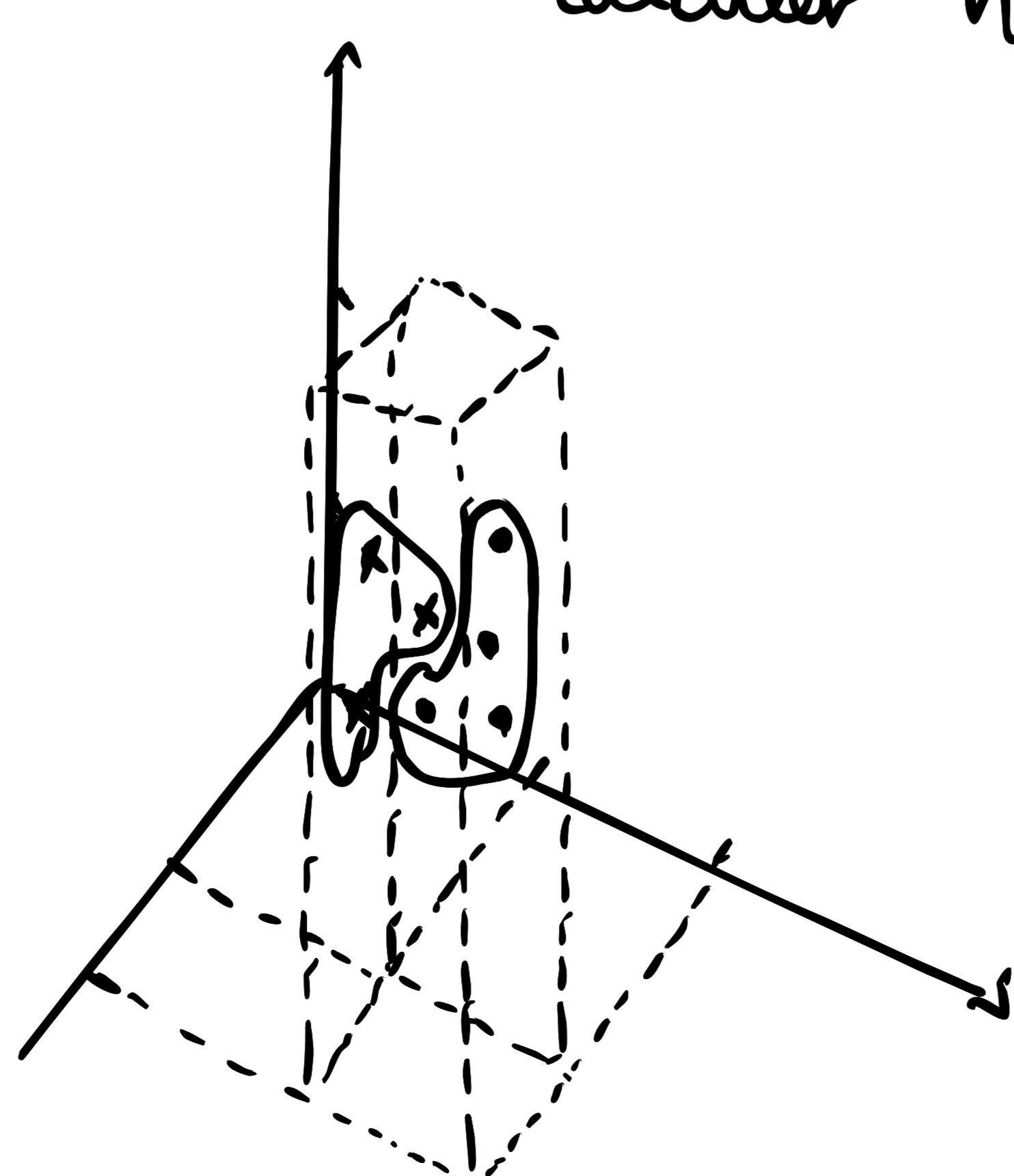
$$\hat{n}(x) = \sum_{k=1}^K \mathbb{1}\{x \in C_k\} \frac{\sum_{i=1}^n \mathbb{1}\{Y_i = +1, x_i \in C_k\}}{\sum_{i=1}^n \mathbb{1}\{x_i \in C_k\}}$$

est l'estimateur de Nadaraya - Watson de la probabilité à postériori.

Autrement dit : • Diviser l'espace en régions

• Dans chaque région, calculer la fréquence

• Calculer  $\hat{n}(x)$



Approche alternative performante, mais dépend grandement du découpage en régions.

• Lissage par kernel pour la classification.

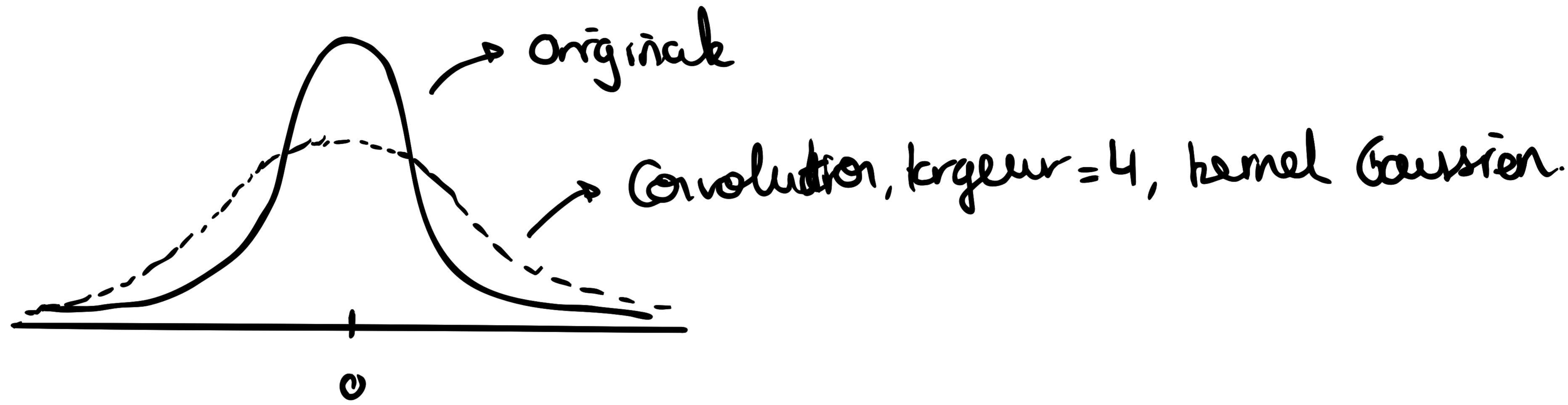
Lisser la frontière de décision.

Remplacer la fonction indicatrice par un kernel de convolution.

$$K: \mathbb{R}^d \rightarrow \mathbb{R}_+, K > 0, \text{ symmetric et } \int K(x) dx = 1$$

Bandé de largeur  $h > 0$  et rescale:

$$K_h(x) = \frac{1}{h} K(x/h).$$



On change alors le classifieur de Nadaraya-Watson :

$$\text{Si } \sum_{i=1}^n \mathbf{1}\{Y_i = 1\} K_h(x - x_i) > \sum_{i=1}^n \mathbf{1}\{Y_i = -1\} K_h(x - x_i)$$

Prédire  $Y = +1$ .

Sinon  $Y = -1$ .

$$\text{La "Plug-in": } \hat{\eta}(x) = \frac{\sum_{i=1}^n \mathbf{1}\{Y_i = +1\} K_h(x - x_i)}{\sum_{i=1}^n K_h(x - x_i)}$$

Si  $\eta$  lisse,  $\hat{\eta}$  peut être meilleur que  $\hat{\eta}$

Dans les KNN, la fonction filtrée est :  $\hat{h}(x) = \sum_{i=1}^n w_i(x) y_i$ , poids  $w_i(x) = \frac{1}{k}$ .  
Avec les kernels  $K$ , fonctions telles que:  $\int K(x) dx = 1$ ,  $\int_x K(x) dx = 0$ .

$$0 < \int x^2 K(x) dx < \infty$$

$$\hat{h}(x) = \frac{\sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) y_i}{\sum_{i=1}^n K\left(\frac{x-x_i}{h}\right)}$$

Mais le kernel souffre d'un biais aux limites du domaine des  $x_i$ .

Possibilité de transformer les  $y_i$  par des polynômes.

### III. Arbres de décision

Comment construire des partitions  $C_1, U \dots C_m$  automatiques d'après les données d'entraînement ?

$$f(x) = \sum_{l=1}^m \mathbb{1}(x \in C_l) (\operatorname{argmax}_c [\sum_{i, x_i \in C_l} \mathbb{1}(y_i = c)])$$

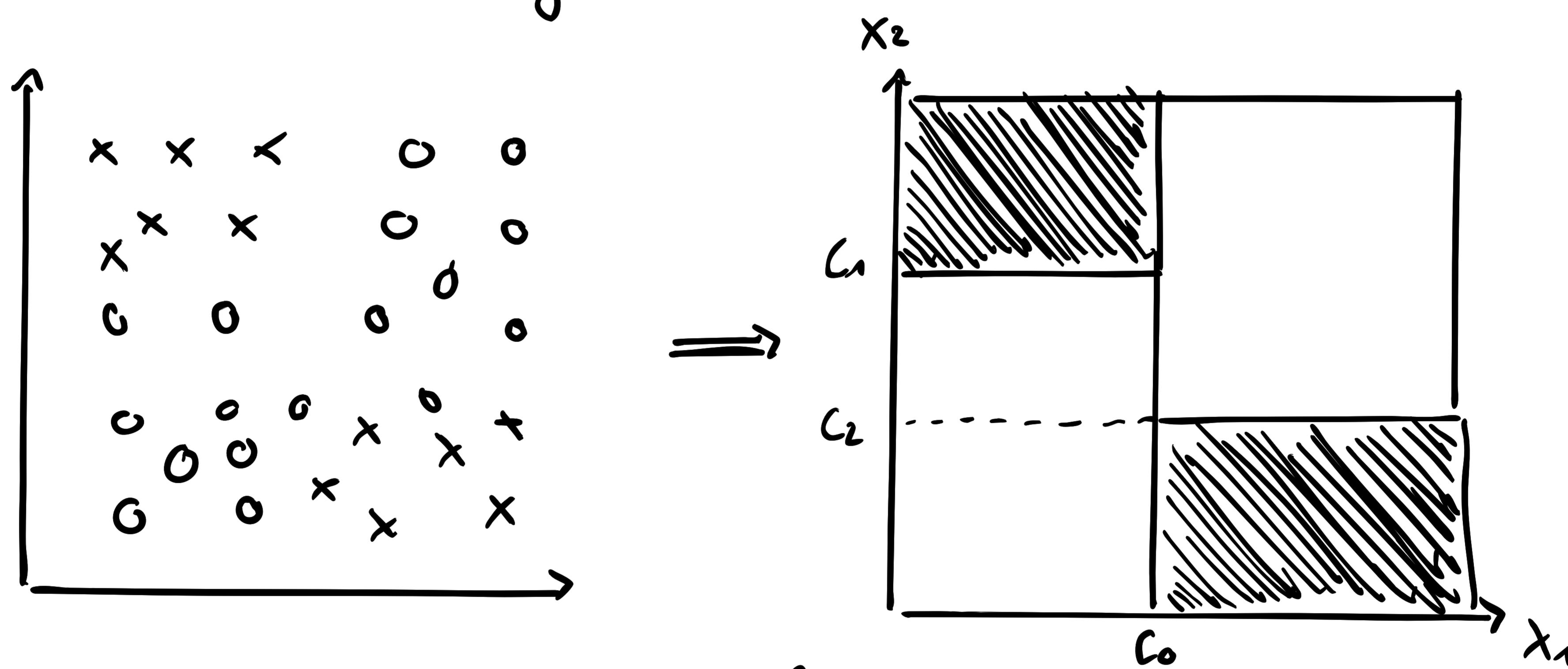
Construire un arbre binaire en choisissant à chaque noeud, une règle de partition qui sépare l'échantillon actuel en deux en minimisant un critère local.

↳ On perte d'algorithme glouton.

↳ Chaque branche de l'arbre devient une feuille et correspond à un subset des données qui prend la valeur du vote majoritaire.

Fonctionne pour:

- les classifications binaires
- les classifications multi-classe
- les régressions

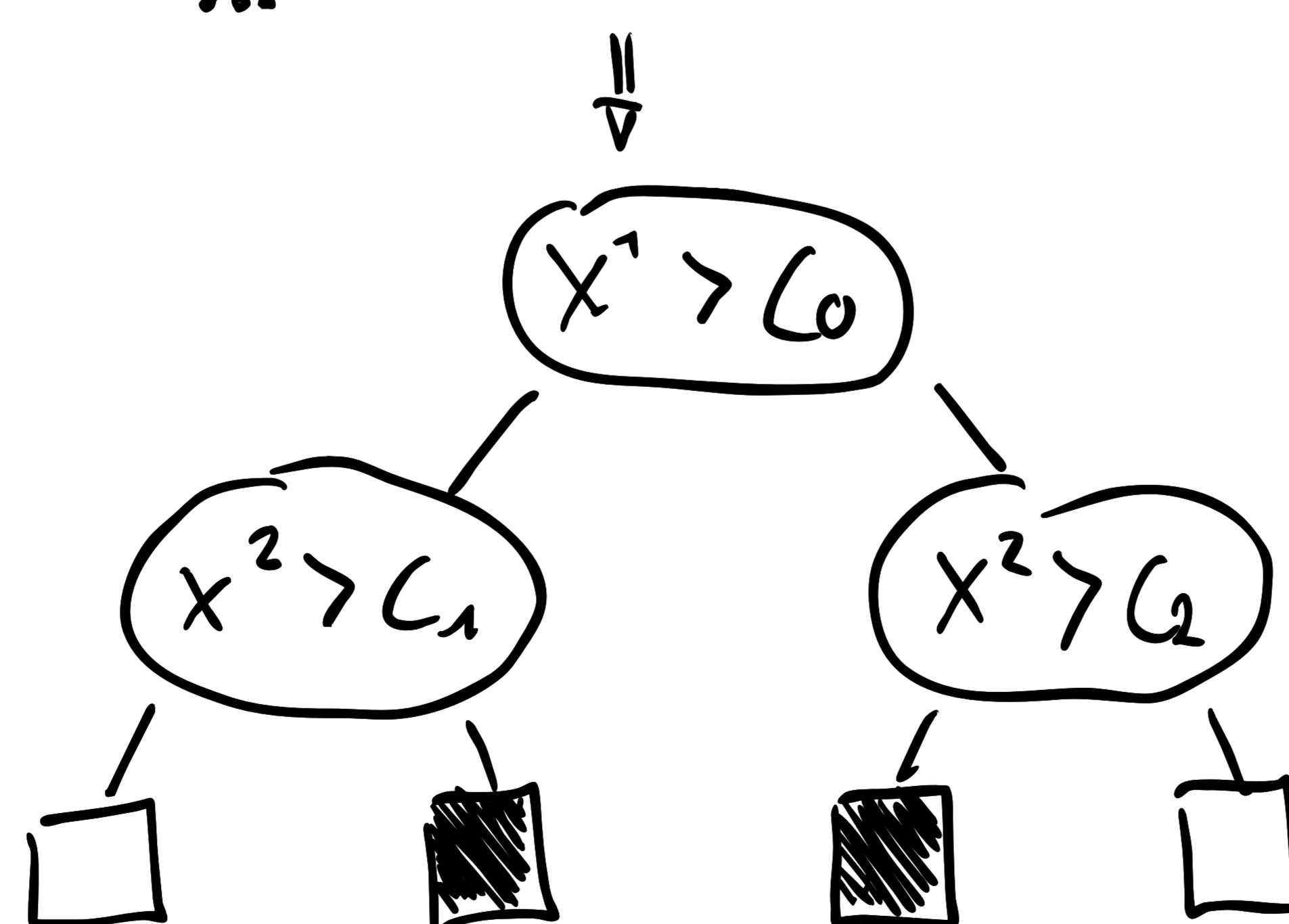


1<sup>re</sup> séparation :  $x^1 > c_0$

2<sup>me</sup> séparation :  $c_1$

3<sup>me</sup> séparation :  $c_2$

...



On utilise non pas 1 mais plusieurs séparateurs linéaires pour contruire les frontières de décision.

- Utiliser des séparateurs linéaires orthogonaux à chaque vecteur de base, i.e des hyperplans de la forme  $x^j = 0$  pour garder une interprétabilité de la fonction à construire.
- À l'issue de la phase d'apprentissage, on connaît les variables explicatives qui internement dans la fonction de décision sont utilisées.

↳ L'arbre code un ensemble de règles logiques du type :

Si  $(x^{j_1} > c_{j_1})$  et  $(x^{j_2} > c_{j_2})$  et ... , alors  $x$  est de la classe  $k$ .

Séparateur linéaire orthogonal à un vecteur de base.

Variable  $x^j$  continue :  $t_{j,c}(x) = \text{sign}(x^j - c)$

Variable  $x^j$  catégorielle à  $K$  valeurs :  $t_{j,v,k}(x) = \mathbb{1}\{x^j = v_k^j\}$ .  
↳  $\{v_1^j, \dots, v_K^j\}$

### Algorithme récursif de construction d'un arbre binaire

- Soit  $S$  l'ensemble d'apprentissage
- Construire un nœud racine
- Chercher la meilleure séparation  $t: X \rightarrow \{0, 1\}$  à appliquer sur  $S$  telle que le coût local  $L(t, S)$  soit minimal.
- Assurer le séparateur choisi au nœud courant et séparer l'ensemble d'apprentissage courant  $S$  en  $S_d$  et  $S_g$  à l'aide de ce séparateur.
- Construire un nœud fils à droite et un nœud à gauche.
- Mesurer le critère d'arrêt à droite, s'il est vérifié, le nœud droit devient une feuille. Sinon, aller en 3 avec  $S_d$  comme ensemble courant.
- Mesurer le critère d'arrêt à gauche, s'il est vérifié, le nœud gauche devient une feuille, sinon aller en 3 avec  $S_g$  comme ensemble courant.

Comment définir le fonction de coût local?

Ensemble d'apprentissage  $S$ , fonction de séparation binaire  $t_{j,\tau}$ .

$$D(S, j, \tau) = \{(x, y) \in S, t_{j,\tau}(x) > 0\}$$

$$G(S, j, \tau) = \{(x, y) \in S, t_{j,\tau}(x) \leq 0\}$$

Parmi tous les paramètres  $(j, \tau) \in \{1, \dots, p\} \times \{\tau_1, \dots, \tau_m\}$ , on cherche  $\hat{j}, \hat{\tau}$  qui minimisent  $L(t_{j,\tau}, S) = \frac{n_d}{n} H(D(S, j, \tau)) + \frac{n_g}{n} H(G(S, j, \tau))$

$$\text{avec } n_d = |D(S, j, \tau)|$$

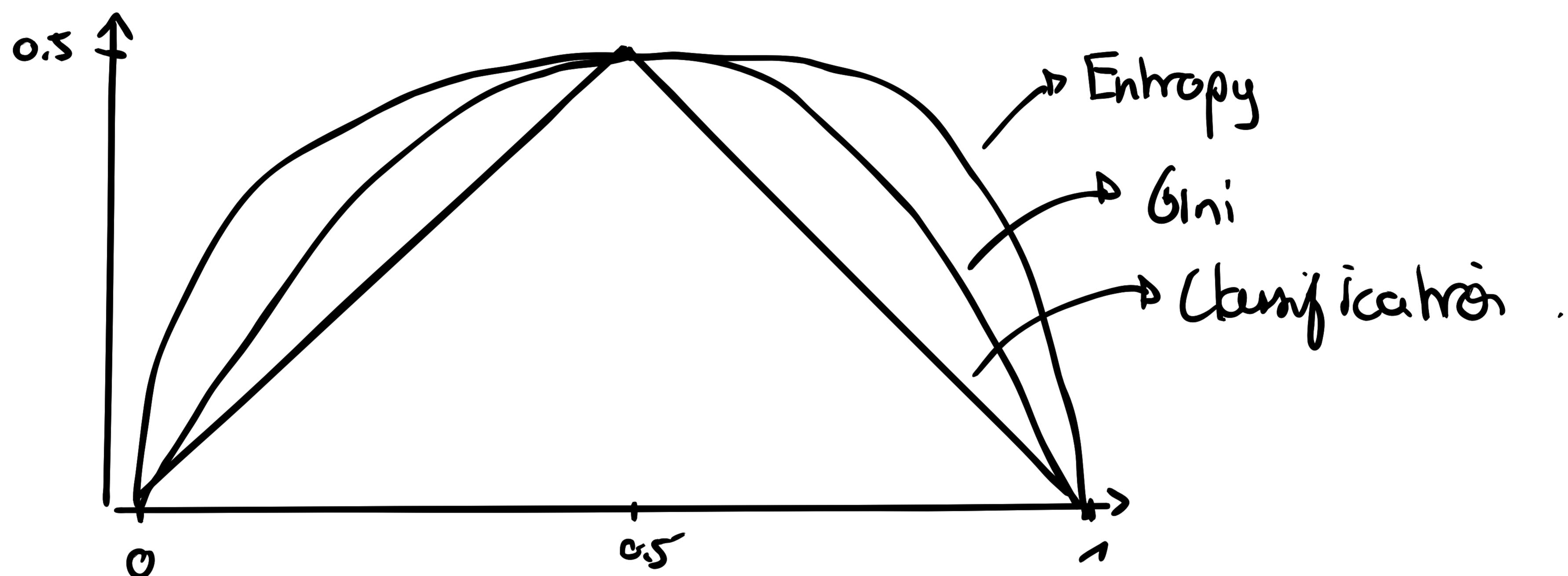
$$n_g = |G(S, j, \tau)|$$

Soit un ensemble  $S$  de  $n$  exemples étiquetés:

$$p_c(S) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}(y_i = c).$$

Quels critères peuvent être utilisés pour  $H$  ?

- Entropie croisée :  $H(S) = - \sum_{l=1}^C p_l(S) \log p_l(S)$
- Index de Gini :  $H(S) = \sum_{l=1}^C p_l(S)(1-p_l(S))$ . (populaire)
- Erreur de classification :  $H(S) = 1 - p_c(S)$ ,  $c(S)$  classe majoritaire dans  $S$ .



Avec  $H$  sélectionné, on peut minimiser la fonction  $L$  et identifier  $j, \tau$ .

Génération des fonctions de séparation candidates:

Pour chaque variable explicative (feature),  $X_j$ :

- $X_j \in [a_j, b_j]$  : définir des seuils réguliers.
- calculer l'histogramme sur les données, prendre des seuils entre les modes.

Dans les problèmes complexes, on peut utiliser deux fois la même variable.

Quels sont les critères d'arrêt ?

"Early stopping" local quand:

- profondeur maximale (populaire)
- nombre de feuilles maximal (moins utilisé)
- nombre minimal d'exemples dans un noeud  $\rightarrow$  On s'arrête si pas assez d'exemples.  
↳ Puis on applique le vote majoritaire dans la fin du noeud.

$\Rightarrow$  Autrement, on apprend jusqu'au bout  $\Rightarrow$  sur-apprentissage.

le critère de la profondeur maximale peut être gênant si l'arbre est asymétrique et qu'un développement profond est souhaitable.

Comment choisir l'hyperparamètre ? Par cross-validation.

NB: Dans chaque noeud, on peut utiliser un ensemble de fonctions séparatrices binaires, mais on perd en interprétabilité. L'arbre est plus compacte par contre.

Autre approche (moins populaire):

laisser l'arbre aller jusqu'au bout, puis élaguer en regardant que les branches qui apportent une amélioration de la performance sur le set de validation.

### Résumé :

Avantages	Désavantages
<ul style="list-style-type: none"><li>Interprétable</li><li>Consistent</li><li>Pas de pre-processing</li><li>Fonctionne avec plusieurs classes</li><li>Fonctionne en prédition <math>O(\log L)</math></li><li>Variables continues ou catégorielles</li></ul>	<ul style="list-style-type: none"><li>Large variance, instable</li><li>Besoin d'ensemble</li><li>Pas d'optimisation globale.</li></ul>

## IV. Arbres de régression

Construire une fonction constante par morceaux, avec la même construction, sauf que le critère local change:

$$L(t_j, \tau, S) = \text{VAR}_{\text{emp}}(S) = \frac{n_d}{n} \text{VAR}_{\text{emp}}(D(j, \tau, S)) - \frac{n_g}{n} \text{VAR}_{\text{emp}}(G(j, \tau, S))$$

Soit  $S$ :  $\text{VAR}_{\text{emp}}(S) = \frac{1}{|S|} \sum_{x_i, y_i \in S} (y_i - \bar{y})^2$ .

On cherche à maximiser l'homogénéité des sorties.

Le problème de régression est :  $l(y_1, y_2) = (y_1 - y_2)^2$

Le meilleur partitionnement est celui tel que la réduction de variance est maximale:

$$\begin{aligned} \text{Score}_r(T_{\text{test}}, S) &= \text{var}\{y | S\} - \frac{N_l}{N} \text{var}\{y_l | S_l\} - \frac{N_r}{N} \text{var}\{y_r | S_r\} \\ &\downarrow \\ &= \frac{1}{N} \sum_{i=1}^N (y_i - \frac{1}{N} \sum_{i=1}^N y_i)^2 \end{aligned}$$

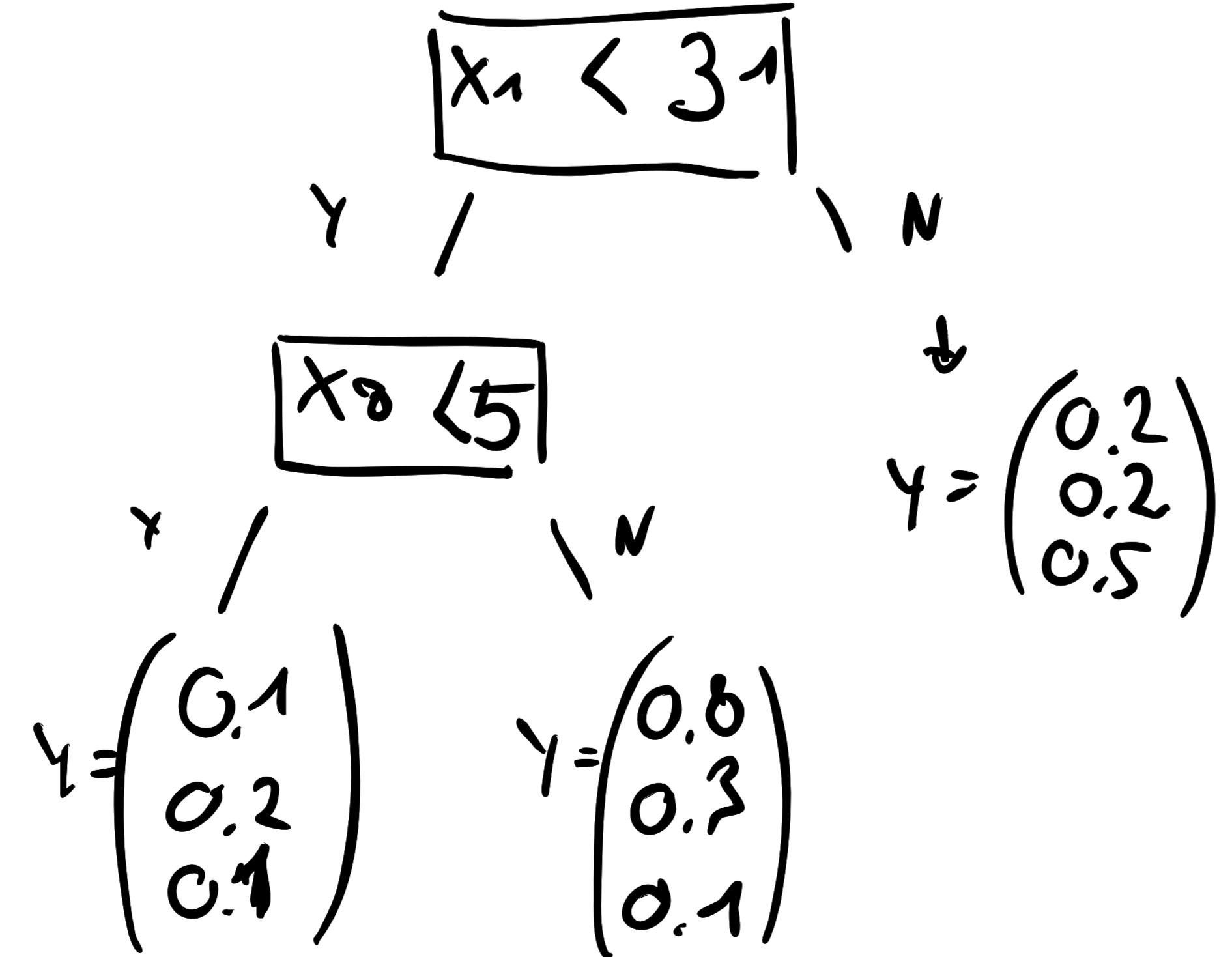
La prédition à un nœud est la moyenne:  $\frac{1}{N_L} \sum_{i=1}^{N_L} y_i$ ,  $N_L$  nb d'ent. dans la feuille.

Un arbre de régression peut être courtisé sur plusieurs nœuds:

$$y = \hat{y}, \quad l(y_1, y_2) = \|y_1 - y_2\|^2.$$

$$\text{var}\{y | S\} = \frac{1}{N} \sum_{i=1}^N \|y_i - \bar{y}\|^2, \quad \bar{y} = \frac{1}{N} \sum_{i=1}^N y_i$$

$t_{\text{tree}}(x) = \frac{1}{N_L} \sum_{i=1}^{N_L} \hat{y}_i \cdot t_i(x)$ ,  $t$  une fonction indicatrice qui donne le nombre de feuilles où  $x$  tombe, et  $N_L$  le nombre de feuilles:



- Robuste
- Interprétable
- Scalable
- Efficient
- Grande variance
- Accuracy faible
- Doit être combiné dans un ensemble.

## Support Vector Machine et méthodes à noyau

### I. Classification binaire supervisée

$X$  vect. a. de  $\mathcal{X} = \mathbb{R}^p$

$y$  var. al. discrète  $y = \{-1, 1\}$

$P$  la loi de proba jointe de  $(X, Y)$ .

$S_n = \{(x_1, y_1), \dots, (x_n, y_n)\}$  sample de  $P$ .

$f: \mathbb{R}^p \rightarrow \{-1, 1\}$  fonction de classification binaire:  $f(x) = \text{sign}(h(x))$

$h: \mathbb{R}^p \rightarrow \mathbb{R} \in \mathcal{H}$

$l: \{-1, 1\} \times \mathbb{R} \rightarrow \mathbb{R}$  fct de perte.

Risque empirique:  $R_n(h) = \frac{1}{n} \sum_i l(y_i, h(x_i))$  et  $\Omega(h)$  mesure la complexité.

↳ On cherche  $\hat{h} = \arg \min_{h \in \mathcal{H}} R_n(h) + \lambda \Omega(h)$

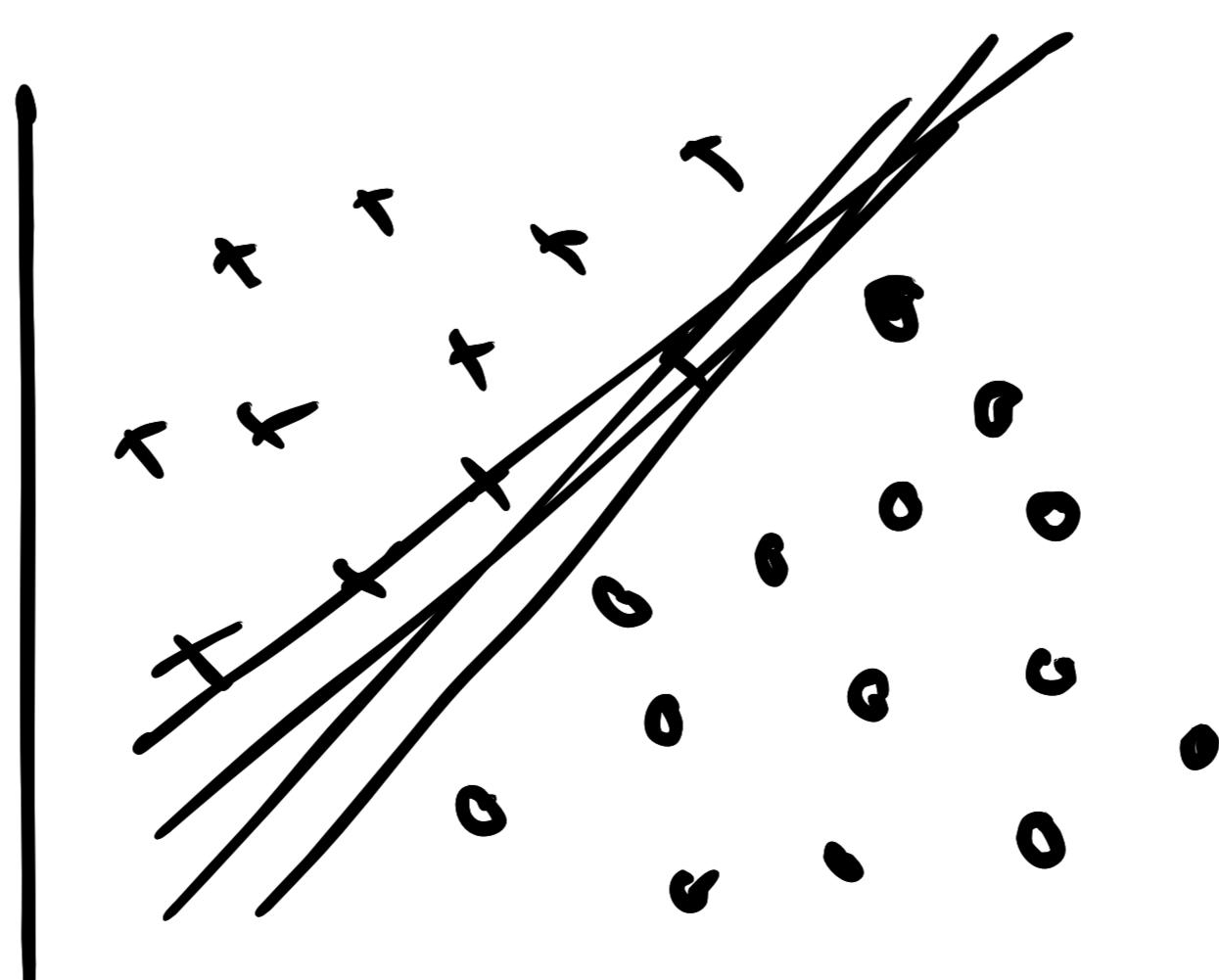
Pour développer une approche discriminante:

- espace de représentation des données
- classe des fonctions de classification binaire considérées
- fonctions de coût à minimiser
- algorithme de minimisation
- méthode de sélection de modèle

#### 1. SVM linéaire

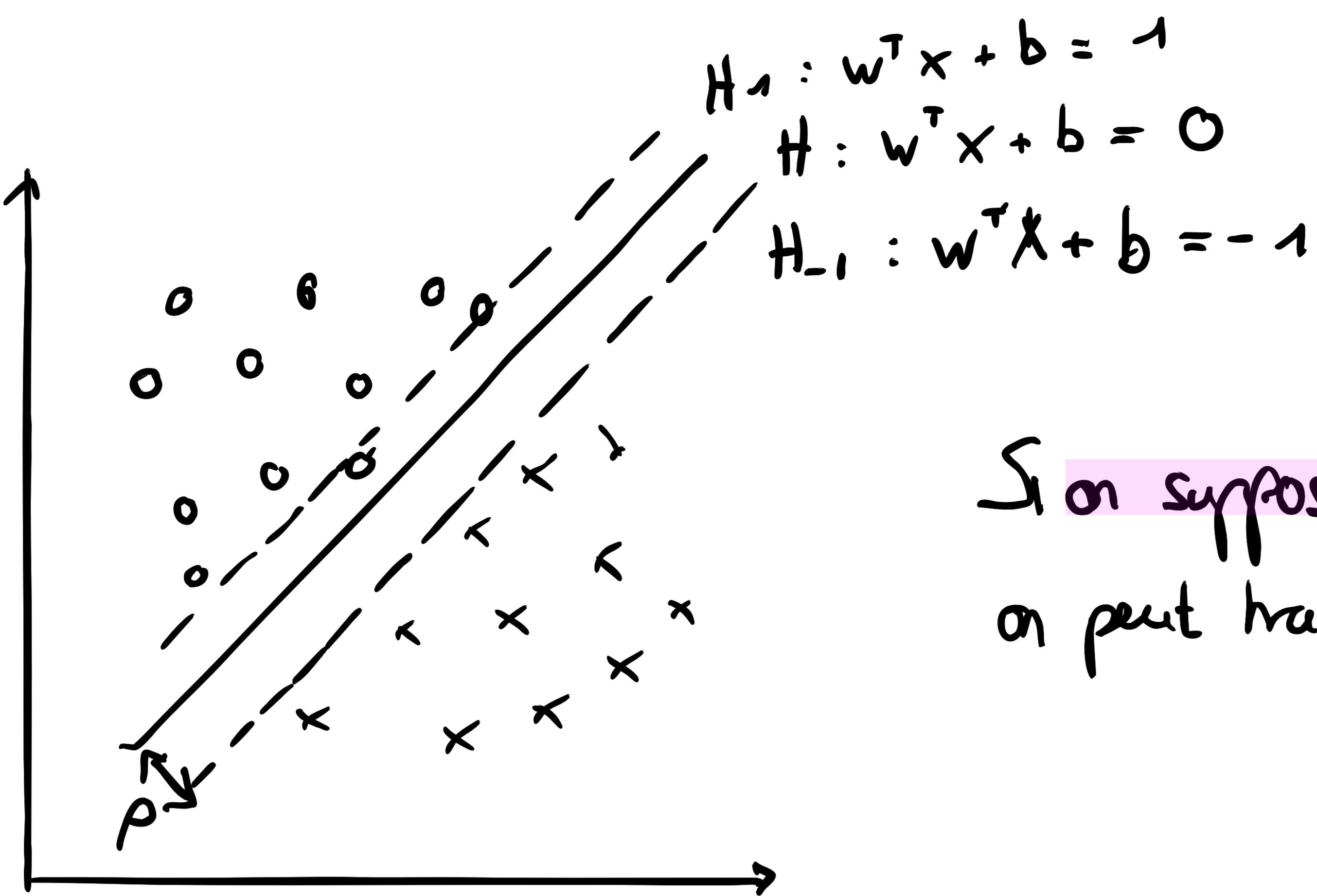
Soit  $x \in \mathbb{R}^p$ ,  $h(x) = \text{sign}(w^T x + b)$ , où  $w^T x + b$  défini un hyperplan dans l'espace euclidien  $\mathbb{R}^p$ .

En 2D par exemple :



Quelle charte choisir ?

↳ On veut maximiser la marge !



Si on suppose les données séparables parfaitement, on peut tracer ces 3 hyperplans.

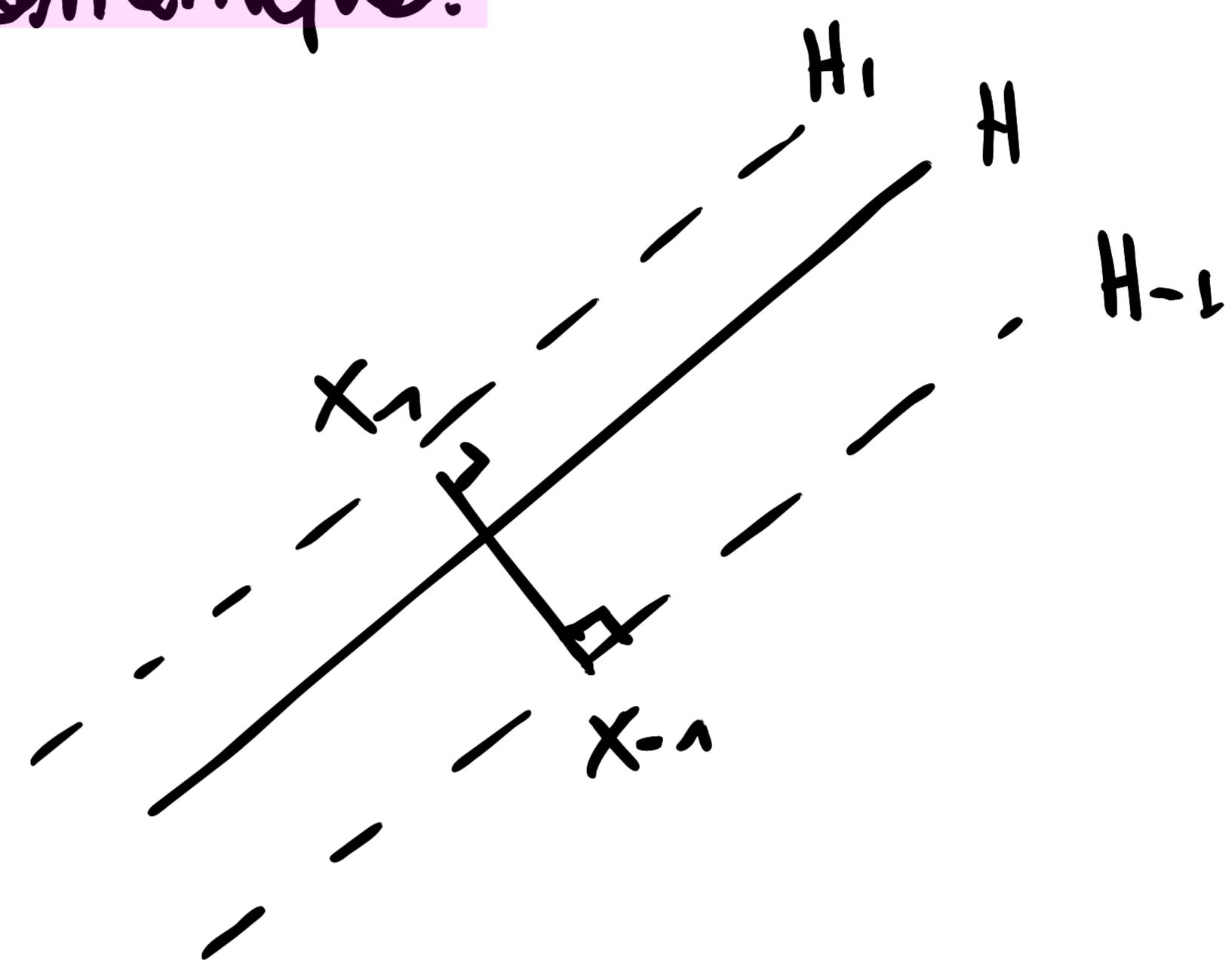
On peut définir la notion de marge géométrique:

- Triplet d'hyperplans:

$$H : w^T x + b = 0$$

$$H_1 : w^T x + b = 1$$

$$H_{-1} : w^T x + b = -1$$



$$\left. \begin{array}{l} \hookrightarrow (x_1 - x_{-1}) = \lambda w \\ x_1 \in H_1 \\ x_{-1} \in H_{-1} \end{array} \right\} \text{Marge géométrique } \rho(w) = \frac{1}{\|w\|} \text{ plus petite distance entre les données et l'hyperplan } H.$$

- Comment déterminer  $w$  et  $b$  ?

Maximiser  $\rho(w)$  tout en séparant les données de part et d'autre de  $H_{-1}$  et  $H_1$ .

$$\text{Données } o : y_i = 1 \rightarrow w^T x_i + b \geq 1$$

$$\text{Données } x : y_i = -1 \rightarrow w^T x_i + b \leq -1$$

### a) Cas séparable

Optimisation dans l'espace primal:

$$\min_{w,b} \frac{1}{2} \|w\|^2 \text{ s.t. } y_i(w^T x_i + b) \geq 1, \quad i=1 \dots n.$$

$$\Leftrightarrow \min_{w,b} \frac{1}{2} \|w\|^2 \text{ s.t. } 1 - y_i(w^T x_i + b) \leq 0, \quad i=1 \dots n$$

C'est une programmation quadratique sous contraintes d'inégalités affines.

$$\Leftrightarrow \text{Lagrangien: } L(w, b, \alpha) = \frac{1}{2} \|w\|^2 + \sum_i \alpha_i (1 - y_i(w^T x_i + b)) \quad \forall i, \alpha_i \geq 0$$

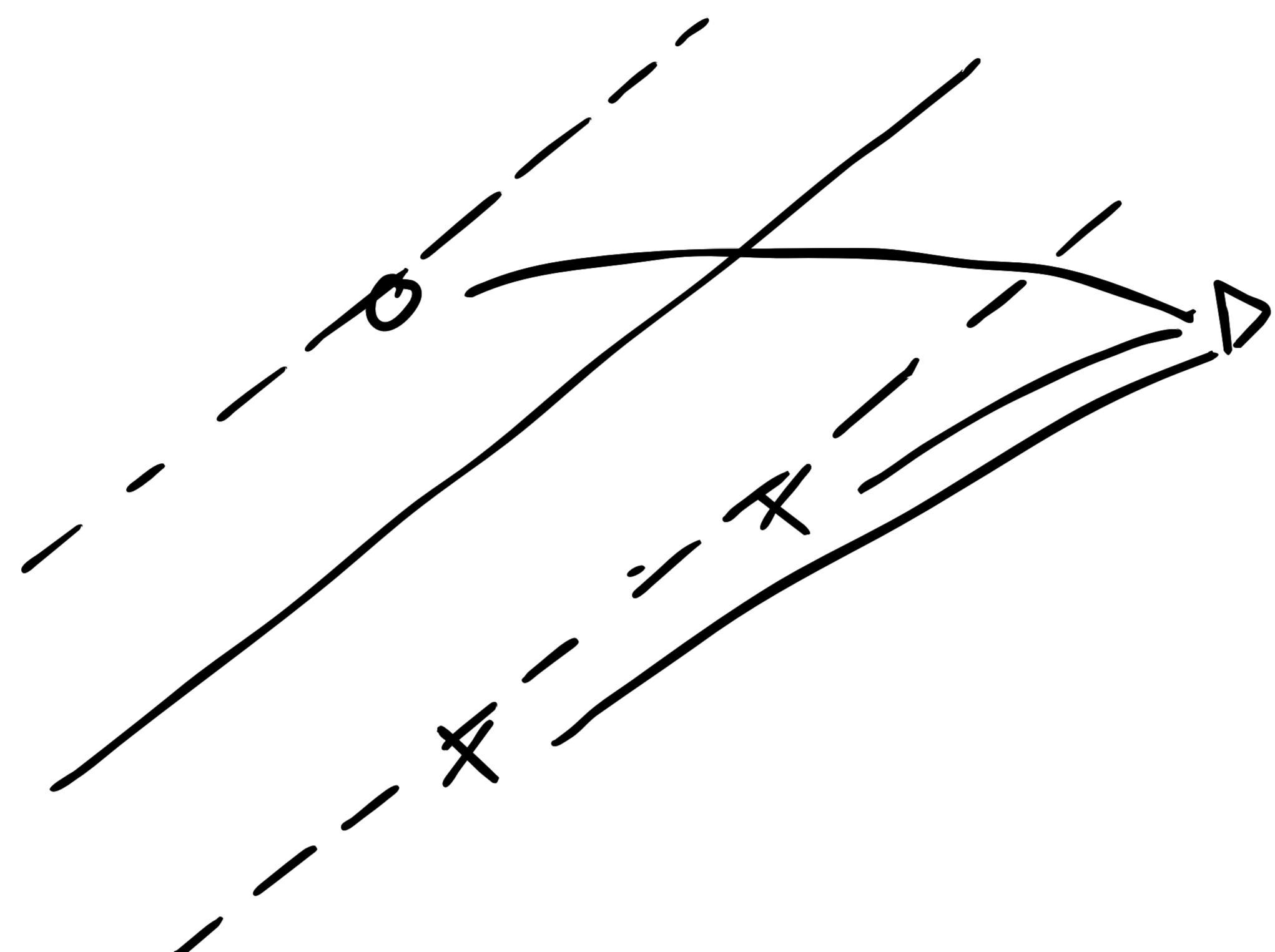
Problème convexe : Point de selle :  $\min_w \max_\alpha L(w, \alpha)$ , et inverse si convexe  $\alpha$ .

Les conditions de Karush - Kuhn - Tucher en l'extrémum sont:

$$\nabla_w \mathcal{L}(w) = w - \sum_{i=1}^n \alpha_i y_i x_i = 0.$$

$$\nabla_b \mathcal{L}(b) = -\sum_{i=1}^n \alpha_i y_i = 0, \quad \alpha_i > 0$$

$$\forall i, \alpha_i [1 - y_i (w^T x_i + b)] = 0$$



Ces 3 données sont des données support.  
Elles servent à déterminer les hyperplans.  
On ne s'intéresse pas aux autres données

Obtenir les  $\alpha_i$ : Résolution dans l'espace dual

En prenant  $w$  et  $b$  les minimiseurs de notre problème

$$\begin{aligned} \mathcal{L}(w, b, \alpha) &= \frac{1}{2} \|w\|^2 + \sum_i \alpha_i (1 - y_i (w^T x_i + b)) \\ &= \frac{1}{2} w^T w + \sum_i \alpha_i - \sum_i \alpha_i w^T x_i - \sum_i \alpha_i y_i b \\ &= \frac{1}{2} (\sum_i (y_i x_i \alpha_i))^T (\sum_i \alpha_i y_i x_i) + \sum_i \alpha_i - \sum_i \alpha_i y_i (\sum_j \alpha_j y_j x_j)^T x_i \\ &\quad - \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j \\ &= -\frac{1}{2} \sum_{i=1, j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_i \alpha_i \quad \text{s.t. } \alpha_i > 0 \end{aligned}$$

$\Leftrightarrow$  Maximiser  $\mathcal{L}$  sous les contraintes  $\alpha_i > 0$  et  $\sum_i \alpha_i y_i = 0 \quad \forall i$ .

On fait appel à un solveur quadratique.

Supposons que les multiplicateurs de lagrange  $\alpha_i$  soient déterminés:

↳ L'équation d'un SVM linéaire est:

$$f(x) = \text{signe} \left( \sum_{i=1}^n \alpha_i y_i x_i^T x + b \right)$$

Pour classer une donnée  $x$ , ce classifieur combine linéairement les valeurs de classe  $y_i$  des données support avec des poids du type  $\alpha_i x_i^T x$ .

### b) Cas non-séparable

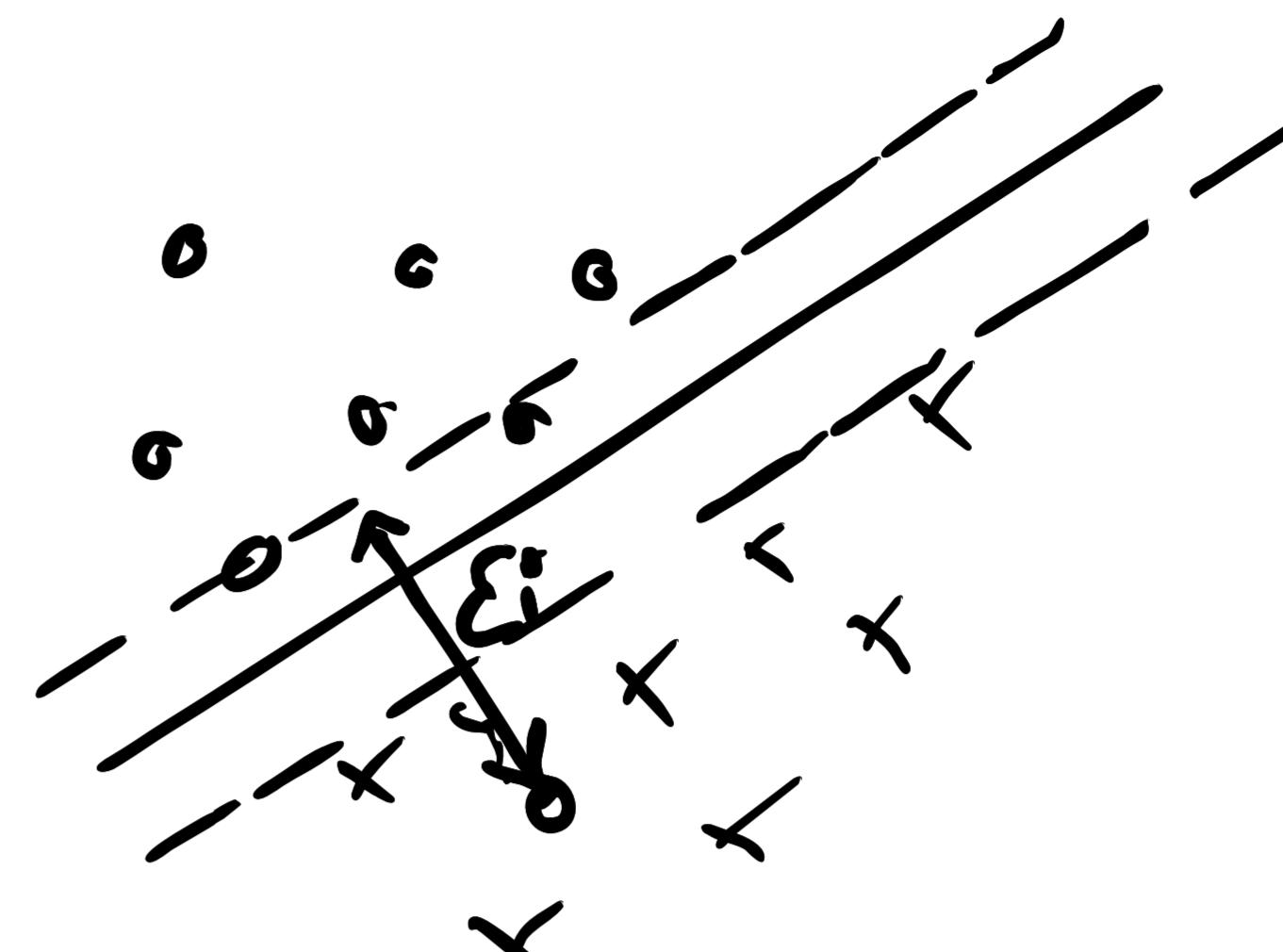
On introduit une variable d'écart  $\varepsilon_i$  pour chaque donnée:

Problème dans le primal:

$$\min_{w, b, \varepsilon} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \varepsilon_i$$

$$\text{s.t. } y_i (w^T x_i + b) \geq 1 - \varepsilon_i, \quad i=1 \dots n$$

$$\varepsilon_i \geq 0$$



On parle alors de **soft-margin** ou marge douce

$$L(w, b, \varepsilon, \alpha, \mu) = \frac{1}{2} w^T w + C \sum \varepsilon_i + \sum_{i=1}^n \alpha_i (1 - \varepsilon_i - y_i (w^T x_i + b)) - \sum \mu_i \varepsilon_i$$

$$\frac{\partial L}{\partial w} = 0 \rightarrow w = \sum_{i=1}^n \alpha_i y_i x_i \quad \text{soit}$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \varepsilon} = C - \alpha_i - \mu_i = 0 \rightarrow \mu_i = C - \alpha_i$$

$$L(\alpha) = \frac{1}{2} \sum_{i=1, j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum \alpha_i + C \cancel{\sum \varepsilon_i} - \sum \mu_i \varepsilon_i$$

$$- \cancel{\sum \alpha_i y_i b} - \sum \mu_i \varepsilon_i$$

$$- C \cancel{\sum \varepsilon_i} + \sum \mu_i \varepsilon_i$$

Problème dans le dual:

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j \quad \text{s.t. } 0 \leq \alpha_i \leq C, \quad i=1 \dots n$$

$$\sum_i \alpha_i y_i = 0$$

On veut un  $C$  relativement grand qui donne beaucoup d'importance au terme (slack variable)  $C \sum_{i=1}^n \varepsilon_i$ .

Soit  $\alpha^*$  la solution du problème dual:

$$\forall i, y_i f_{w^*, b^*}(x_i) - 1 + \varepsilon_i^* \leq 0$$

$$\alpha_i^* \geq 0$$

$$\alpha_i^* [y_i f_{w^*, b^*}(x_i) - 1 + \varepsilon_i^*] = 0$$

$$\mu_i^* \geq 0$$

$$\mu_i^* \varepsilon_i^* = 0$$

$$\alpha_i^* + \mu_i^* = C$$

$$\varepsilon_i^* \geq 0$$

$$w^* = \sum_i \alpha_i^* y_i x_i$$

$$\sum_i \alpha_i^* y_i = 0.$$

Soit  $\lambda^*$  la solution du problème dual:

- Si  $\alpha_i^* = 0$ ,  $\mu_i^* = C > 0$  et donc  $\varepsilon_i^* = 0$ .

$x_i^*$  est bien classé mais n'intervient pas dans le paramétrage de la fonction de décision  $\rightarrow$  Pas d'erreur, et pas sur l'hyperplan séparateur.

- Si  $0 < \alpha_i^* < C$ , alors  $\mu_i^* > 0$  et donc  $\varepsilon_i^* = 0$ .  $x_i^*$  est sur un des hyperplans  $H_1$  ou  $H_2$  est support  $\rightarrow$  Pas d'erreur, sur la frontière.
- Si  $\alpha_i^* = C$ ,  $\mu_i^* = 0$ ,  $\varepsilon_i^* = 1 - y_i f_{w^*, b^*}(x_i)$ ,  $x_i^*$  est support. (mal class.)

On calcule  $b^*$  en utilisant un  $i$  tq  $0 < \alpha_i^* < C$ .

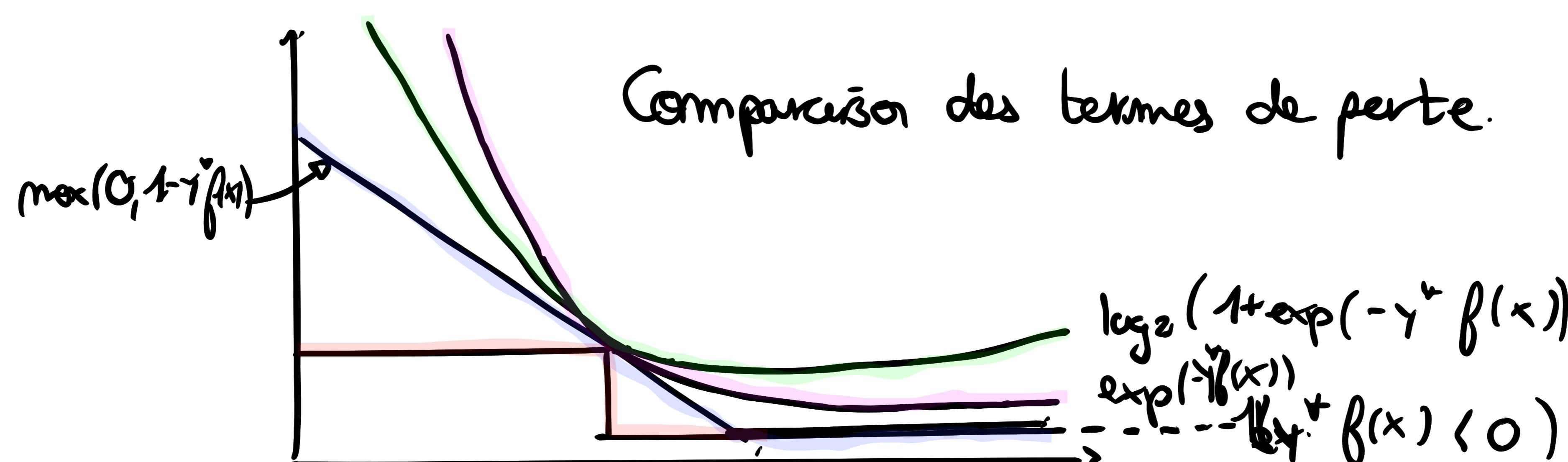
Si  $0 < \alpha_i < C$ : ( $S = \{i \text{ tq } 0 < \alpha_i < C\}$ )

$$1 - y_i f(x_i) = 0$$

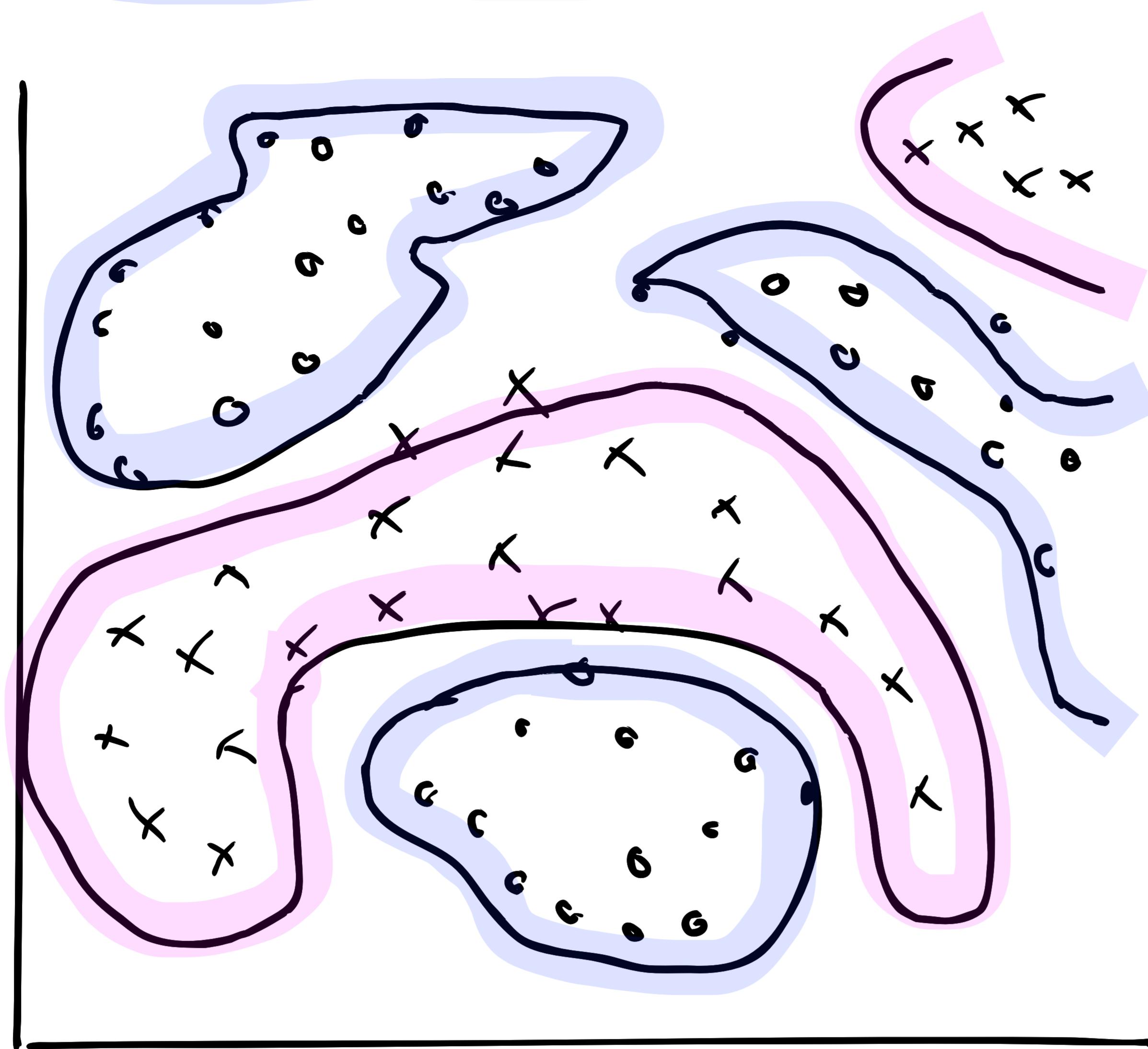
$$1 - y_i (w^T x_i + b^*) = 0.$$

$$b^* = \frac{-1 + y_i w^T x_i}{y_i}$$

$$\tilde{b} = \frac{1}{|S|} \sum_{i \in S} b_i$$



## 2. SVM non-linéaire



Intuition des noyaux :

$$f(x) = \text{signe} \left( \sum_i \alpha_i y_i k(x_i, x) + b \right)$$

au lieu de  $x_i^T x$ .

$$\text{où } k(x, x') = \exp(-\gamma \|x - x'\|^2)$$

NB:

Le problème de l'hyperplan ( séparateur ) de marge optimale ne fait intervenir les données d'apprentissage qu'à travers des produits scalaires.

$$\max_{\alpha} \sum_i d_i - \frac{1}{2} \sum_{i,j} d_i d_j y_i y_j x_i^T x_j$$

$$\text{s.c. } 0 < \alpha_i < C, \sum_i d_i y_i$$

Si on transforme les données à l'aide d'une fonction  $\phi$  non-linéaire, et qu'on pose le problème dans l'espace primal:

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

$$\text{s.c. } y_i (w^T \phi(x_i) + b) \geq 1 - \xi_i, \quad i=1 \dots n$$

$$\xi_i \geq 0, \quad i=1 \dots n.$$

Alors on peut apprendre une fonction de séparation non-linéaire.

► Pour classer une nouvelle donnée  $x$ , on a seulement besoin de savoir calculer  $\phi(x_i)^T \phi(x)$ :  $f(x) = \text{sign} \left( \sum_i \alpha_i y_i \phi(x_i)^T \phi(x) + b \right)$

Au lieu de définir  $\phi$ , on remplace  $x_i^T x_j$  par l'image par une fonction  $k$ :  $k(x_i, x_j)$  telle qu'il existe un espace de re-décriptions (feature space)  $\mathcal{F}$  et une fonction de re-décriptions (feature map)  $\phi: X \rightarrow \mathcal{F}$  et

$\forall (x, x') \in X$ ,  $k(x, x') = \phi(x)^T \phi(x')$ , et alors on peut appliquer le même algorithme d'optimisation (résolu dans le dual).

On obtient :  $f(x) = \text{sign}(\sum_{i=1}^n y_i k(x_i, x) + b)$ .

où  $k$  existe et est appelé noyau.

### a) Définition des noyaux.

Soit  $X$  un ensemble,  $k: X \times X \rightarrow \mathbb{R}$  une fonction symétrique. La fonction " $k$ " est appelée noyau positif défini ssi quelque soit le sous-ensemble fini  $\{x_1, \dots, x_m\}$  de  $X$  et le vecteur colonne  $c$  de  $\mathbb{R}^m$ :

$$\hookrightarrow c^T K c = \sum_{i,j=1}^m c_i c_j k(x_i, x_j) \geq 0$$

$$K: m \times n$$

$$k_{ij} = k(x_i, x_j), \quad x_i \in X$$

### • Théorème de Moore - Aronzajn

Soit  $K$  un noyau positif défini. Alors il existe un espace d'Hilbert  $\mathcal{F}$ , appelé espace de redescription, et une fonction  $\phi: X \rightarrow \mathcal{F}$  appelée fonction de redescription (feature map) tq :  $\langle \phi(x), \phi(x') \rangle_{\mathcal{F}} = k(x, x')$

$\hookrightarrow$  NB : Pour un noyau, il peut exister plusieurs couples  $(\mathcal{F}, \phi)$

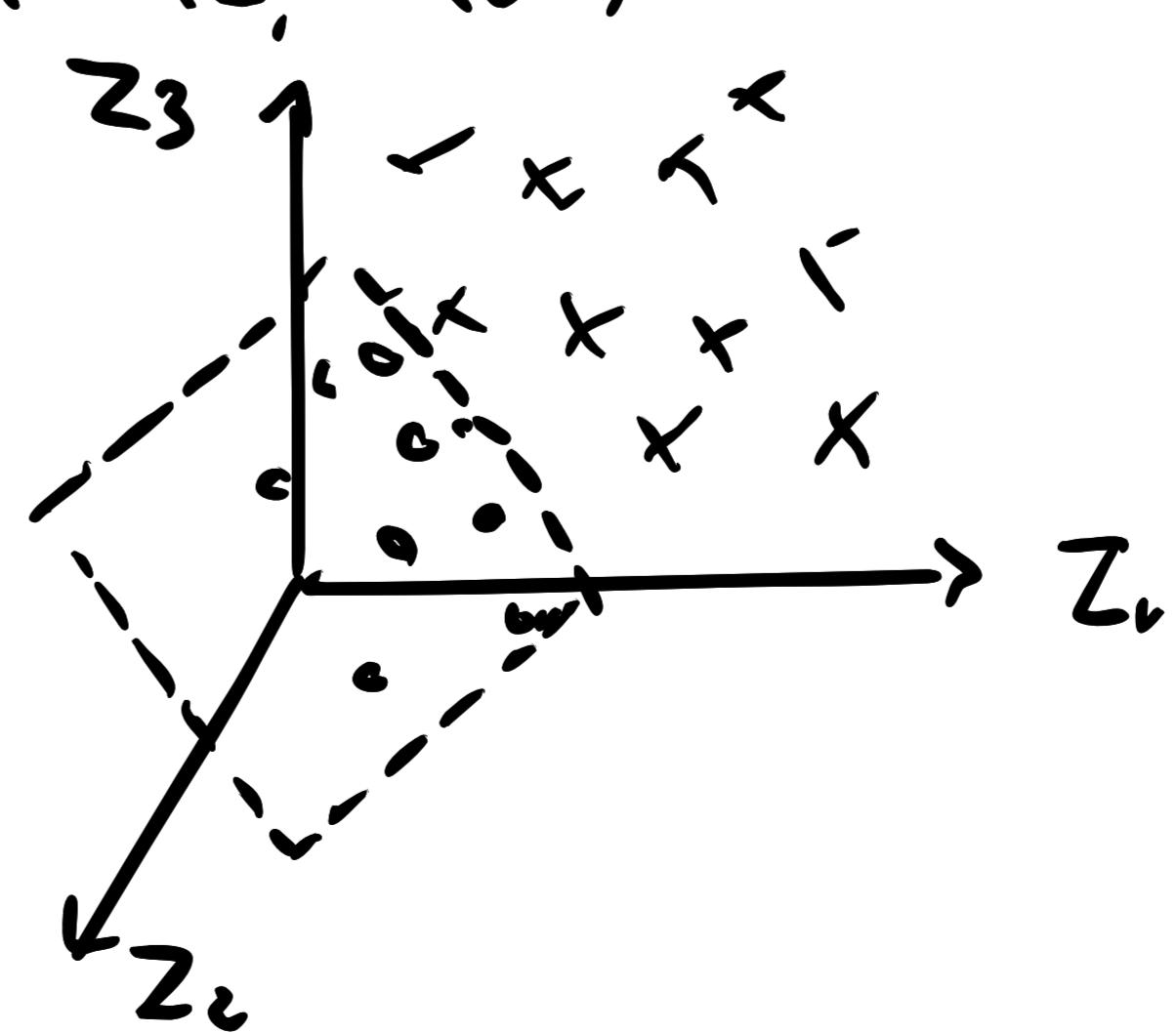
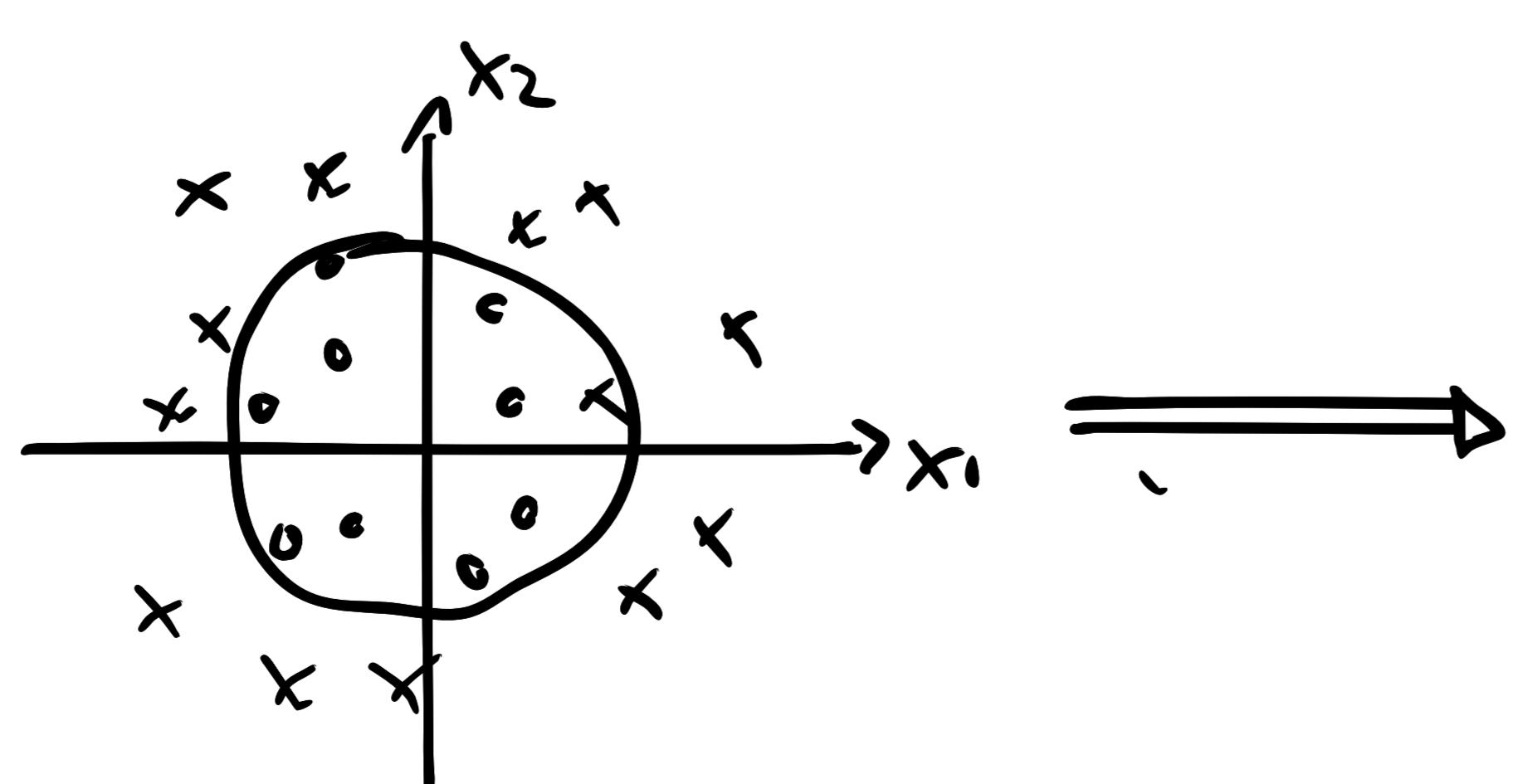
Noyaux entre vecteurs :  $\forall x, x' \in \mathbb{R}^p$

- Noyau linéaire :  $k(x, x') = x^T x'$  (produit scalaire)
- Noyau polynomial :  $k(x, x') = (x^T x' + c)^d$  (ajout d'une constante  $c$ )
- Noyau gaussien :  $k(x, x') = \exp(-\gamma \|x - x'\|^2)$

ex: Noyau polynomial:

$$\Phi: \mathbb{R}^2 \rightarrow \mathbb{R}^3$$

$$(x_1, x_2) \mapsto (1, z_2, z_3) := (x_1^2, \sqrt{2} x_1 x_2, x_2^2)$$



$$\Phi \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1, x_2 \\ x_2^2 \end{bmatrix}$$

$$\phi(x)^\top \phi(x') = x_1^2 x_1'^2 + 2x_1 x_2 x_1' x_2' + x_2^2 x_2'^2 \\ = (x^\top x')^2$$

$\Leftrightarrow k(x^\top x') = (x^\top x')^2$  (Astuce du noyau)

Pourquoi passer par  $k$  plutôt que par  $\phi$  ?

- Le noyau gaussien est très puissant. C'est comme si on travaillait dans des espaces de dimensions infini.
- La famille de modèles gaussien est suffisamment riche pour approximer presque tous les problèmes.
- il y a une algèbre associée aux noyaux :

### Propriétés de fermeture:

Soyons  $k_1, k_2$  deux noyaux sur  $X \times X$ ,  $g: X \rightarrow \mathbb{R}$ ,  $a \in \mathbb{R}$ :

$$h(x, x') = k_1(x, x') + k_2(x, x')$$

$$k(x, x') = a k_1(x, x')$$

$$k(x, x') = g(x) g(x')$$

$$k(x, x') = k_1(g(x), g(x'))$$

Autres usages des noyaux pour des données structurées :

- graphes
- séquences
- arbres
- appliquer les SVM.

ex : - classifier des molécules

- classifier des documents structurés
- traiter des séquences biologiques.

On applique donc les noyaux dans le cas des SVM :

$$f(x) = \text{sign} \left( \sum_{i=1}^n \alpha_i y_i k(x_i, x) + b \right)$$

!! En plus du paramètre  $C$ , il faudra aussi régler la valeur du paramètre

$\gamma$ . Dans chacun des  $\gamma$  (allant de 0.1 à 5 par exemple), on trouve un minimum différent pour  $C$ .

Comment définir un noyau pour une application spécifique ?

- Utiliser les propriétés de fermeture pour construire de nouveaux noyaux.
- les noyaux peuvent être utilisés pour comparer différents types de données
- Apprentissage de noyaux
  - apprentissage d'hyperparamètres
  - apprentissage de noyau multiple : étant donné  $h_1, \dots, h_m$ , apprendre une combinaison convexe :  $\sum_i \beta_i h_i$  de noyaux.

## II. Classification Multiclasse.

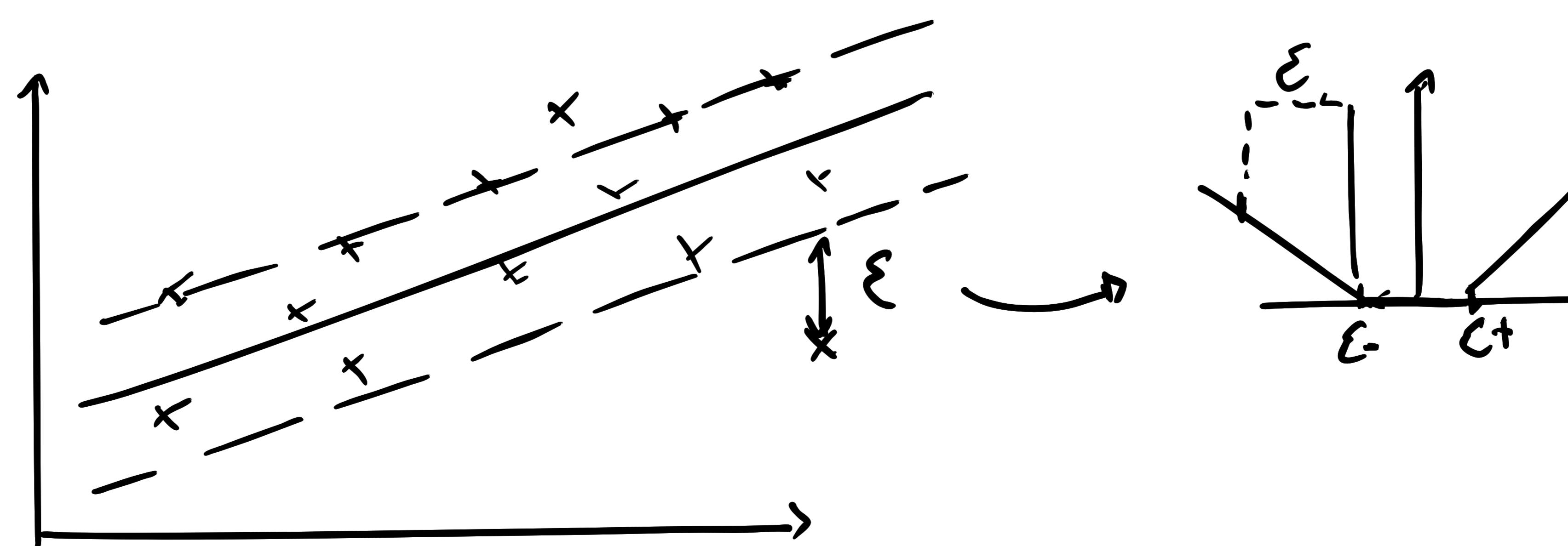
Les SVM multiclasses sont peu efficaces.

En cas multiclasse, il est préféré d'effectuer  $k$  SVM binaires.

## III. Régretton

Supposons  $S_{\text{Scop}} = \{(x_i, y_i), i=1 \dots n\}$  un échantillon iid tiré de la loi  $P$ .

à partir de  $S_{\text{Scop}}$ , déterminer  $f \in \mathcal{F}$  qui minimise  $R(f) = \mathbb{E}_P[\ell(x, y, f(x))]$   
Il est une fonction de coût local. Mesure à quel point la vraie cible et la prédiction par le classifieur sont différentes



On impose un  $\varepsilon$ -tube. Partie  $\varepsilon$ -ménable:  $|y' - y|_\epsilon = \max(0, |y' - y| - \varepsilon)$

SVR en espace primal:

Sachant  $C$  et  $\epsilon$

$$\min_{w, b, \xi} \frac{1}{2} \|w\|^2 + C \sum_i (\xi_i + \xi_i^*)$$

$$\text{s.c. } y_i - f(x_i) \leq \varepsilon + \xi_i$$

$$f(x_i) - y_i \leq \varepsilon + \xi_i^*$$

$$\xi_i^* \geq 0$$

$$\text{avec } f(x) = w^\top \phi(x) + b.$$

Cas général:  $\phi$  est un feature map associé à un noyau défini positif.

En espace dual:

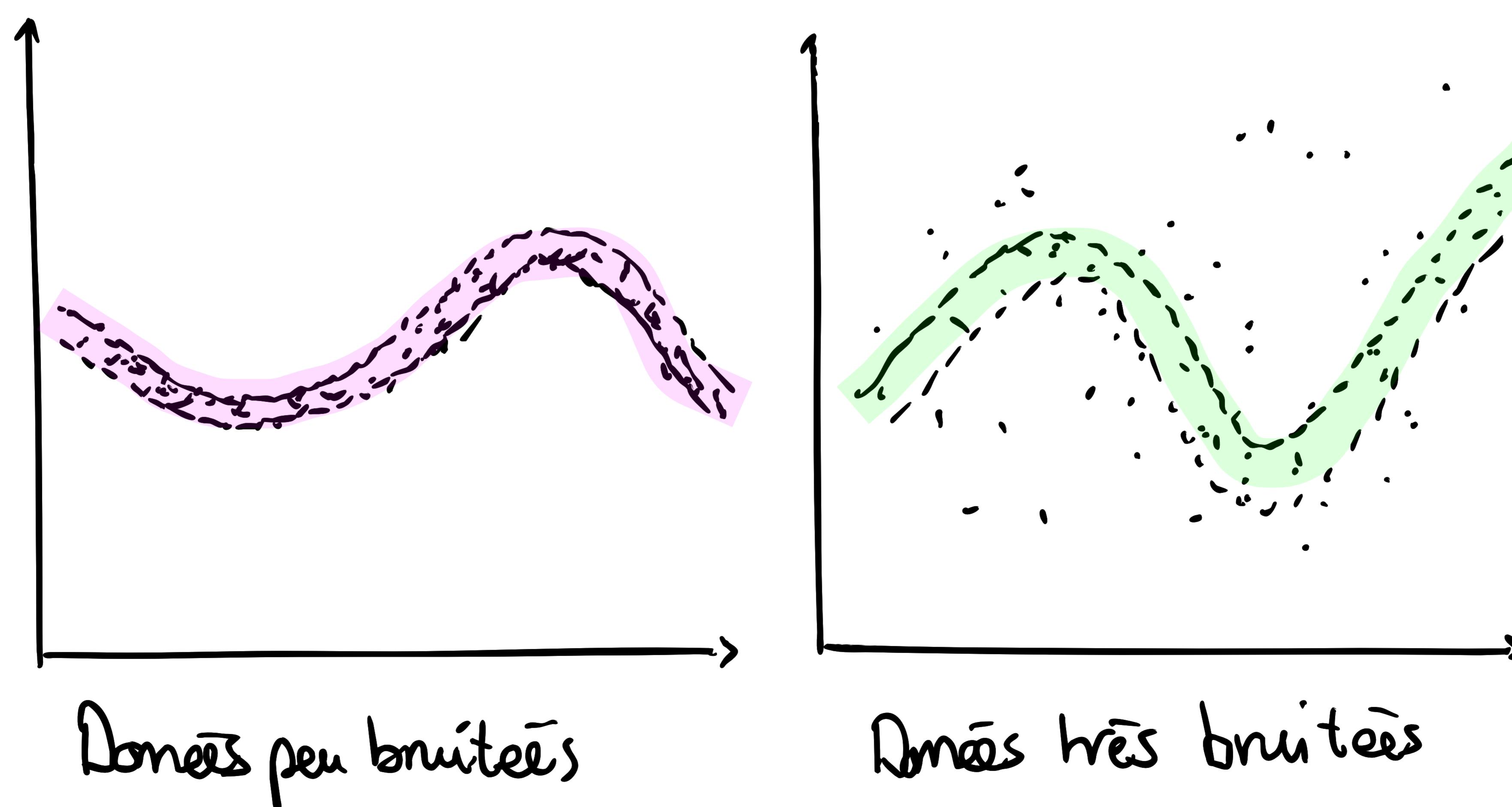
$$\min_{\alpha, \alpha^*} \sum_{i,j} (d_i - \alpha_i^*)(d_j - \alpha_j^*) k(x_i, x_j) + \epsilon \sum_i (\alpha_i + \alpha_i^*) - \sum_i y_i (\alpha_i - \alpha_i^*)$$

$$\text{s.c. } \sum_i (\alpha_i - \alpha_i^*) = 0$$

$$0 \leq \alpha_i \leq C,$$

$$0 \leq \alpha_i^* \leq C.$$

$$\hookrightarrow f(x) = \sum_{i=1}^n (d_i - \alpha_i^*) k(x_i, x) + b$$



## Conclusion :

### Avantages:

- Minimum unique
- Noyau universel (Gaussien pex.)
- Flexible selon noyau
- Multi classe
- Inspiré de Vapnik / Chervonenkis

### Désavantages:

- Choix du noyau
- Algo d'optimisation coûteux en temps et en mémoire  $\rightarrow$  programmation quadratique
- Pour passer à l'échelle:
  - approximations de Nyström (Gram)
  - Random Fourier features (approx spectrale du noyau)

### Astuce du noyau:

- s'applique à d'autres algorithmes
- PCA  $\rightarrow$  Kernel PCA
- CCA  $\rightarrow$  Kernel CCA
- s'applique en sortie:
  - traiter des fonctions à sorties complexes et non à valeurs réelles
  - requiert d'utiliser en entrée des noyaux à valeurs opérateurs