INSTITUT
de MATHEMATIQUES
de TOULOUSE

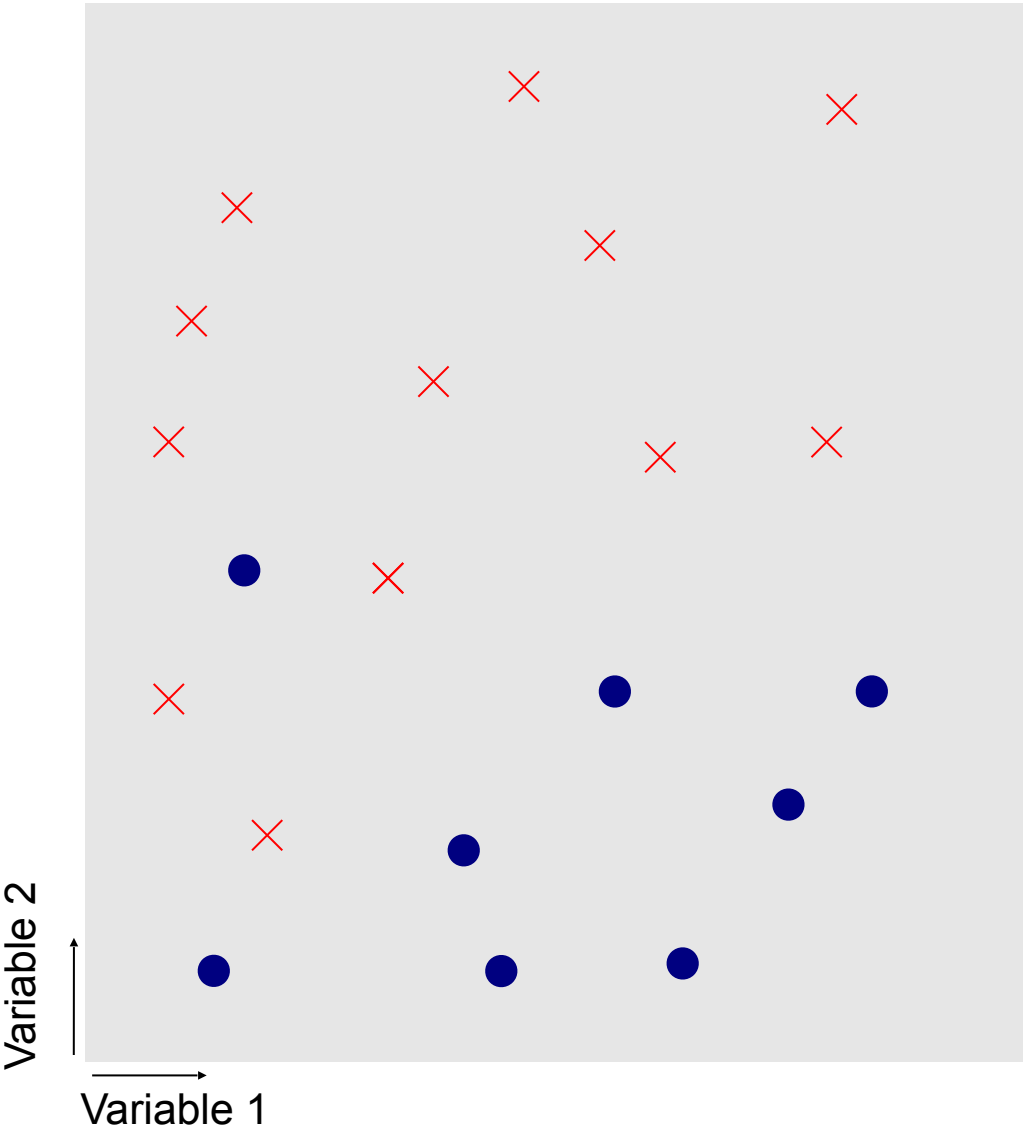
Introduction à l'apprentissage machine

Laurent Risser
Ingénieur de Recherche à l'Institut de Mathématiques de Toulouse

lrisser@math.univ-toulouse.fr

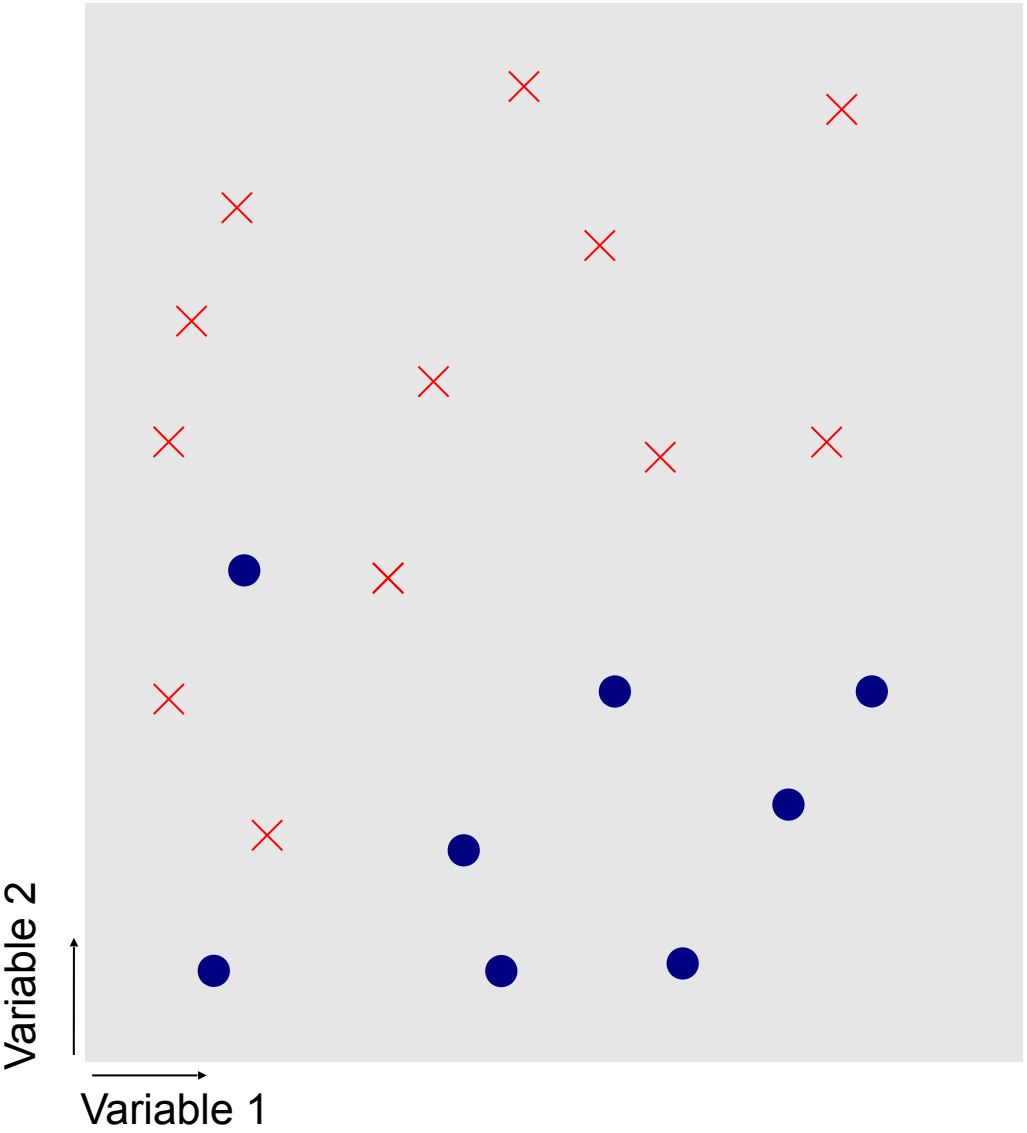
Exemples introductifs à l'apprentissage machine

Exemple introductif 1 : Apprentissage supervisé — *Jeu de données*



$n = 20$ observations
$p = 2$ variables (dimension)
Label à 2 états

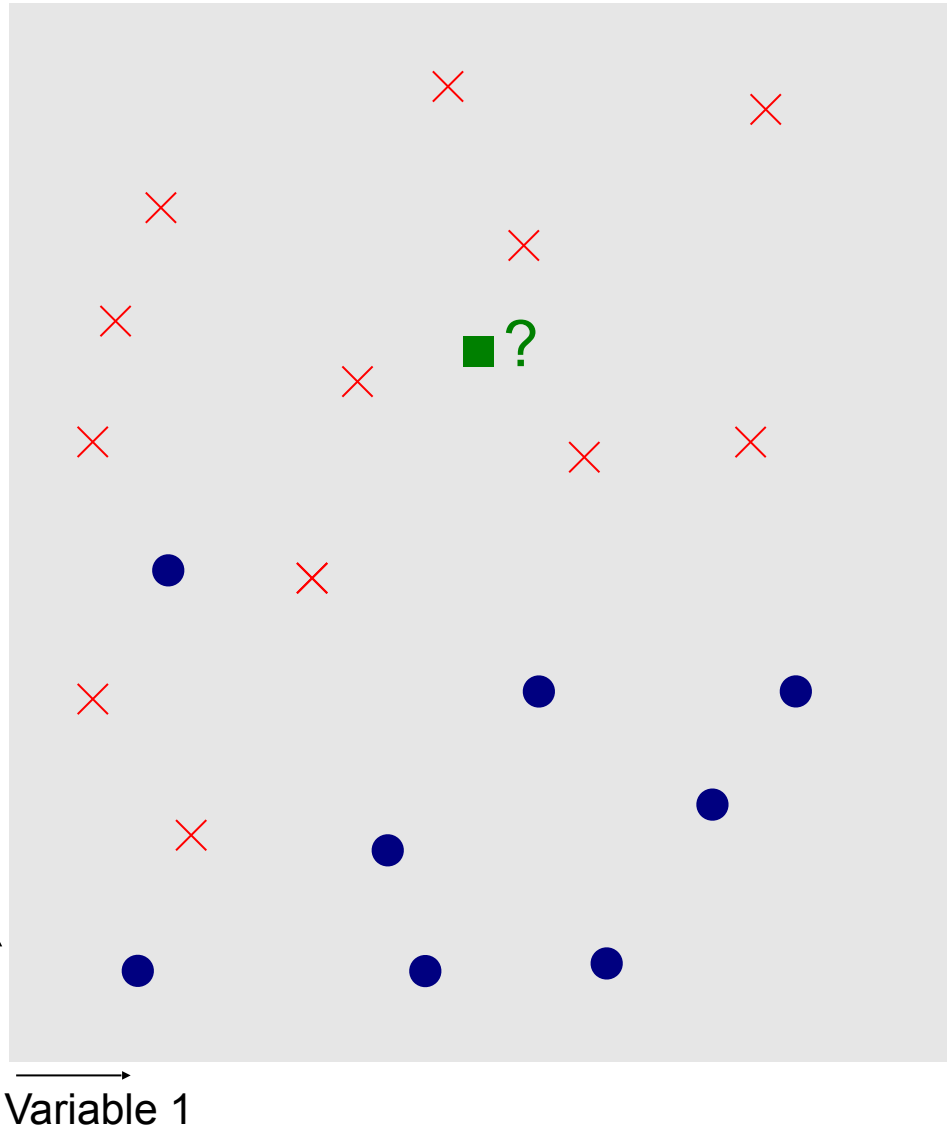
Exemple introductif 1 : Apprentissage supervisé — *Jeu de données*



n = *20* observations
p = *2* variables (dimension)
Label à 2 états

Exemple :
Variable 1 = Age
Variable 2 = Revenu annuel
Etat = Achat d'un type de produit

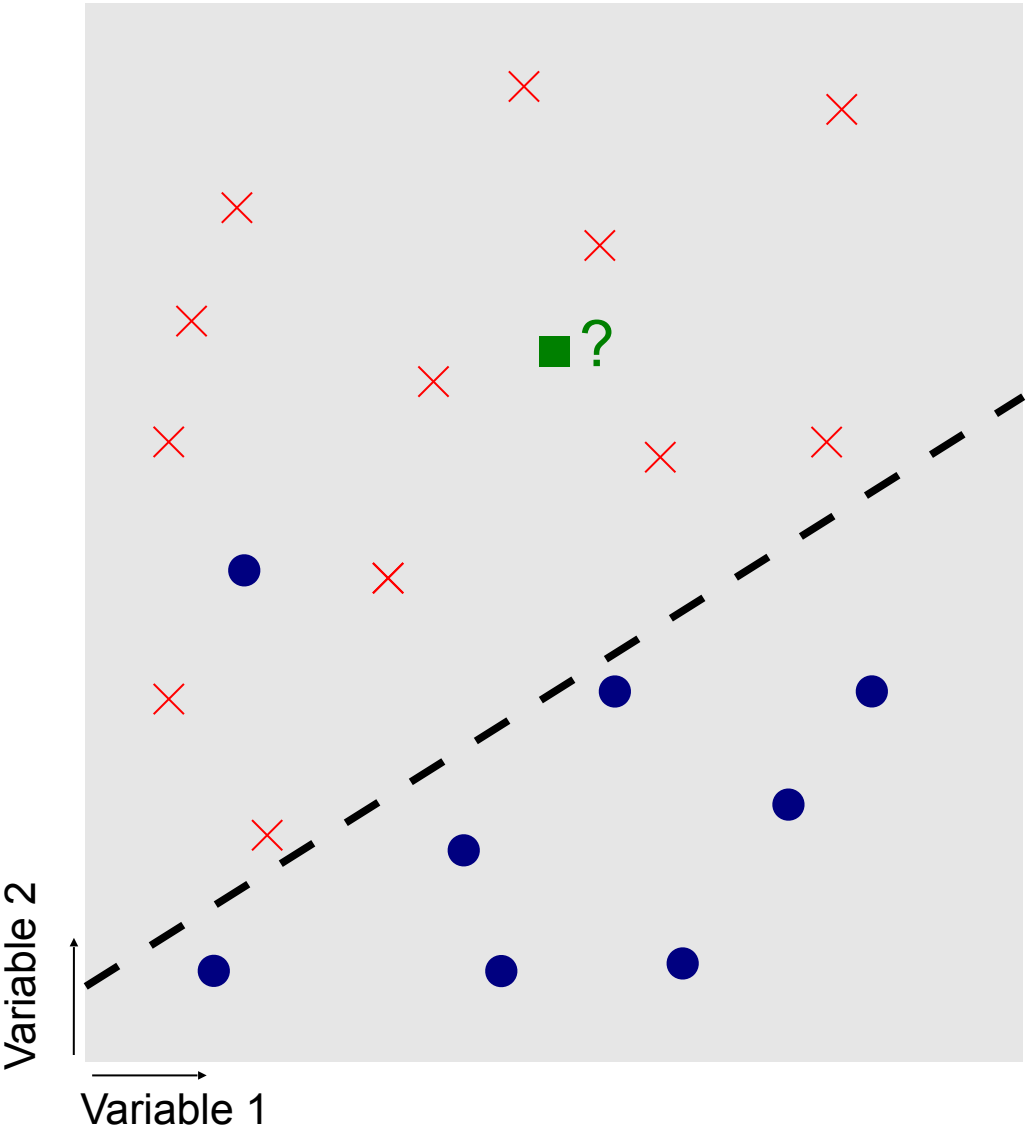
Exemple introductif 1 : Apprentissage supervisé — *Prédiction*



$n = 20$ observations
$p = 2$ variables (dimension)
Label à 2 états

Etat le plus vraisemblable de ? ■

Exemple introductif 1 : Apprentissage supervisé — *Apprentissage*

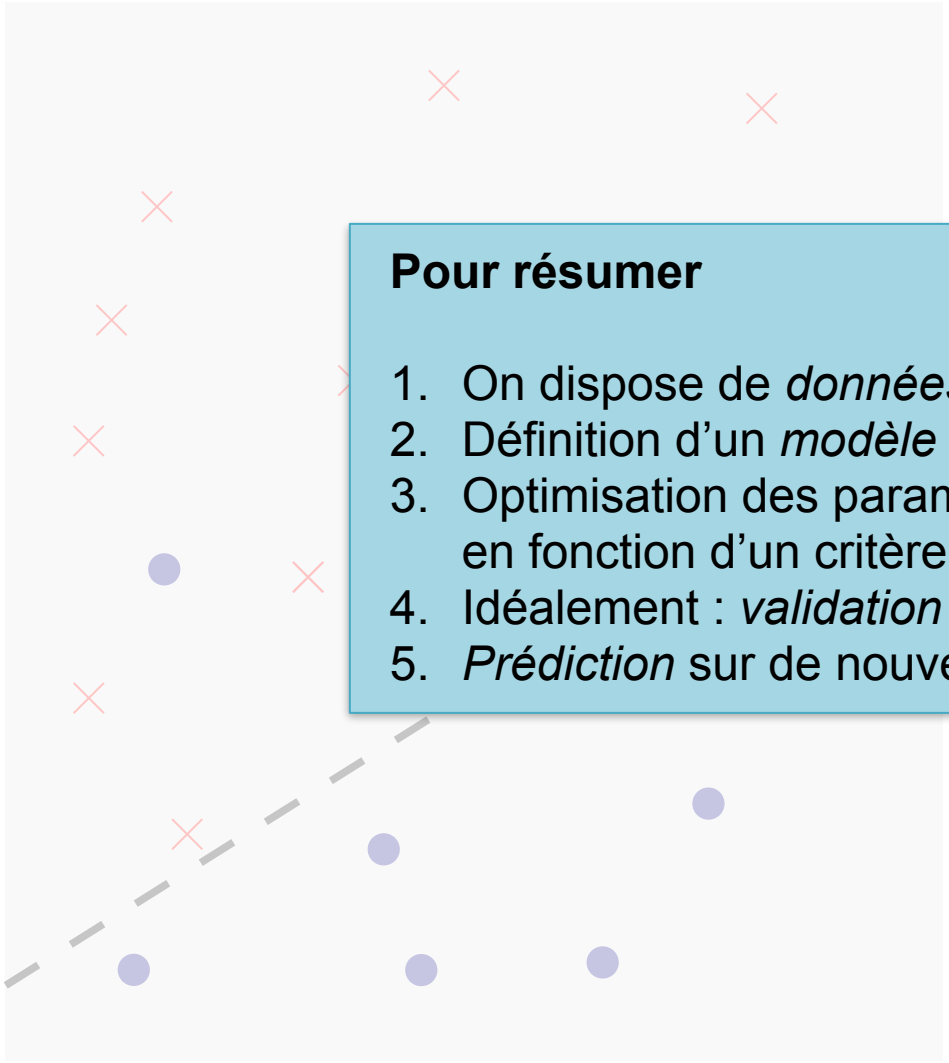


n = *20* observations
p = *2* variables (dimension)
Label à 2 états

Etat le plus vraisemblable de ? ■

⟶ Apprentissage (ici par un modèle linéaire) puis décision/estimation

Exemple introductif 1 : Apprentissage supervisé — *Apprentissage*



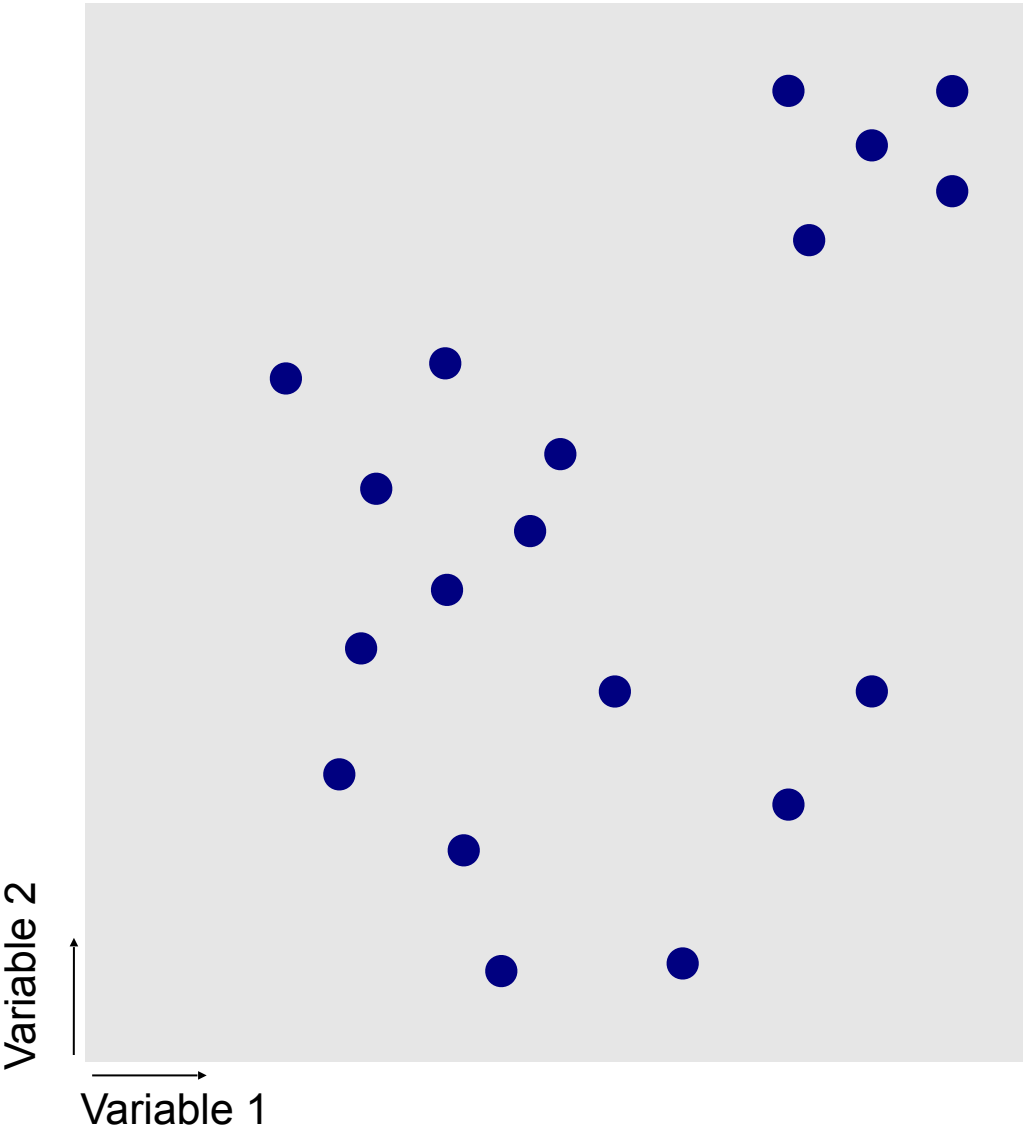
Pour résumer

1. On dispose de *données d'apprentissage* ***annotées***
2. Définition d'un *modèle* pour séparer les données
3. Optimisation des paramètres du modèle (*apprentissage*) en fonction d'un critère d'**erreur de prédiction**
4. Idéalement : *validation* du modèle sur données test
5. *Prédiction* sur de nouvelles observation

Etat le plus vraisemblable de ?

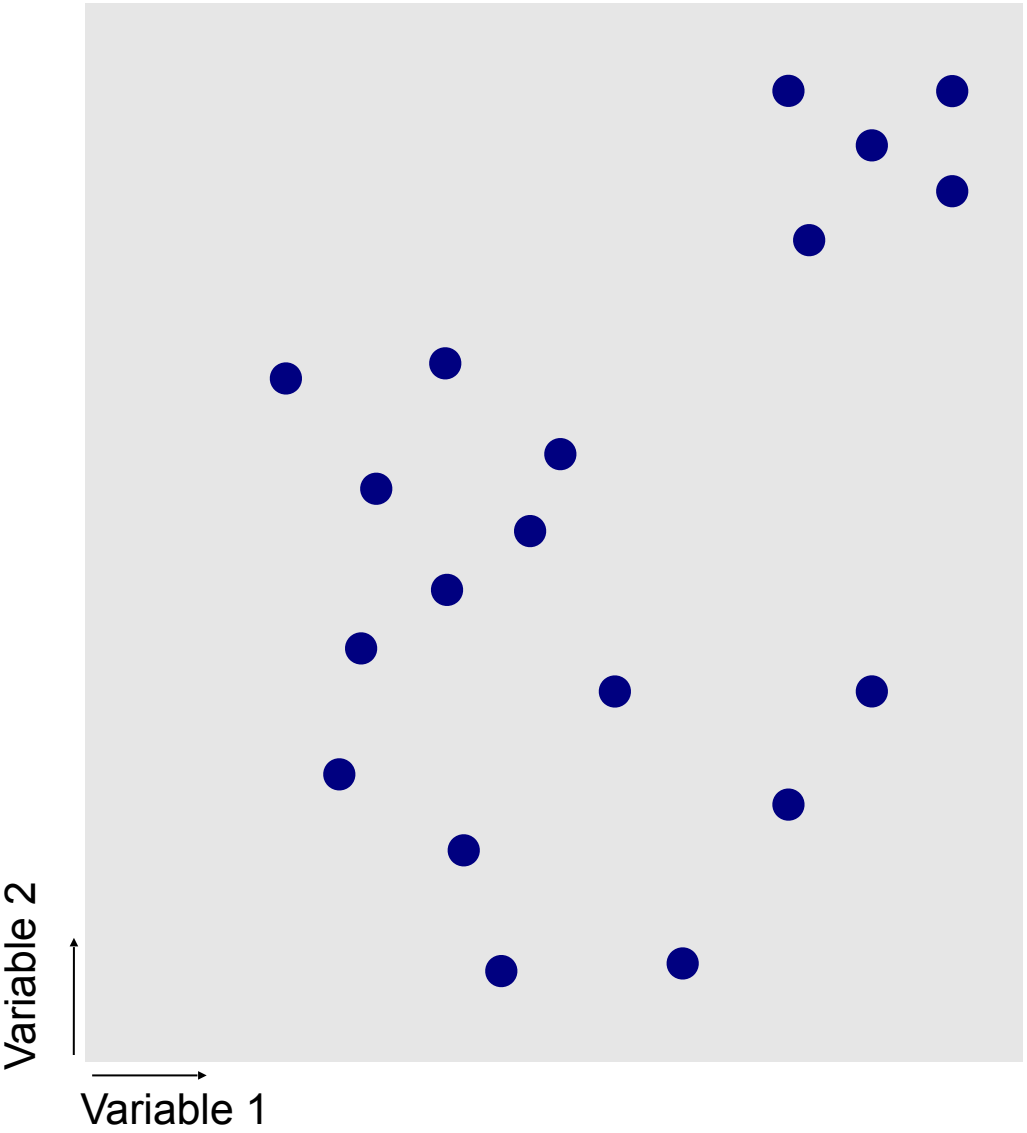
→ Apprentissage par un modèle linéaire puis décision/estimation

Exemple introductif 2 : Apprentissage non-supervisé — *Jeu de données*



$n = 20$ observations
$p = 2$ variables (dimension)
Pas de label

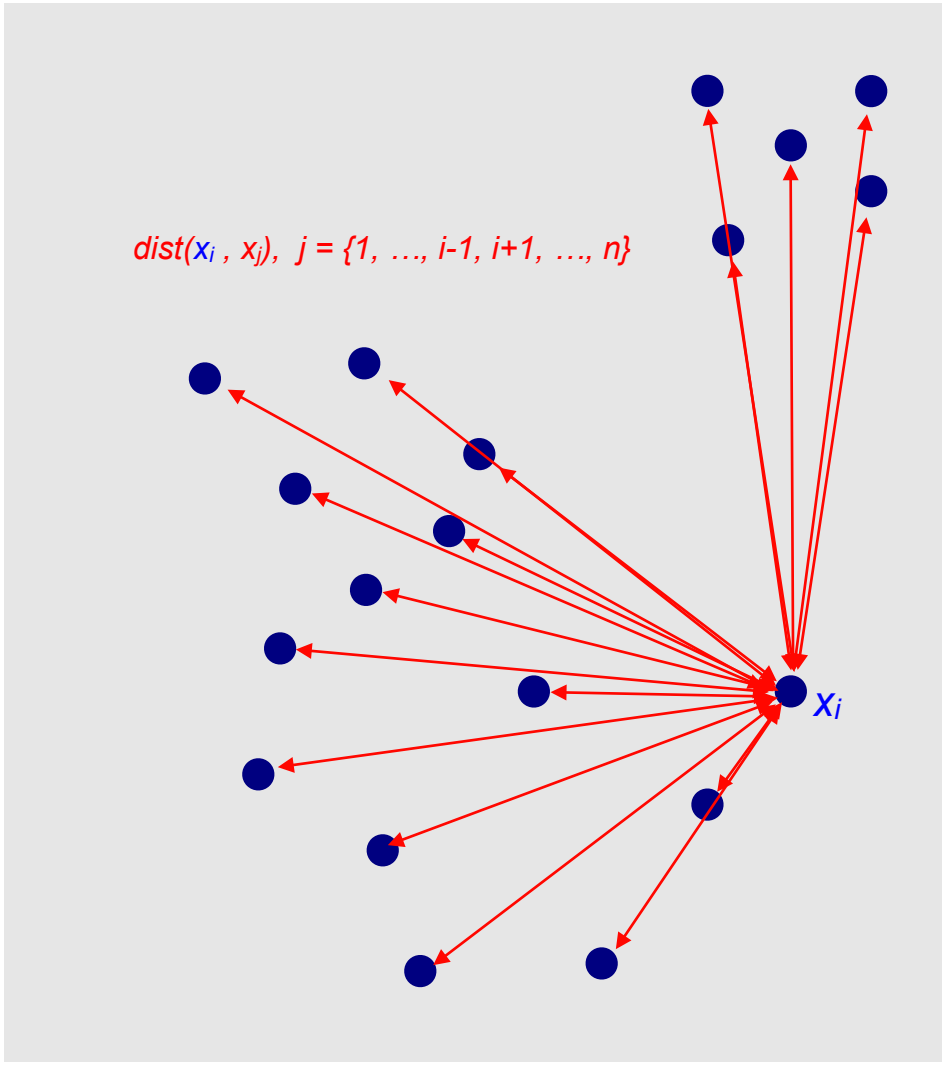
Exemple introductif 2 : Apprentissage non-supervisé — *Jeu de données*



$n = 20$ observations
$p = 2$ variables (dimension)
Pas de label

Peut-on raisonnablement distinguer plusieurs sous-groupes d'individus ?

Exemple introductif 2 : Apprentissage non-supervisé — *Distance entre les observations*

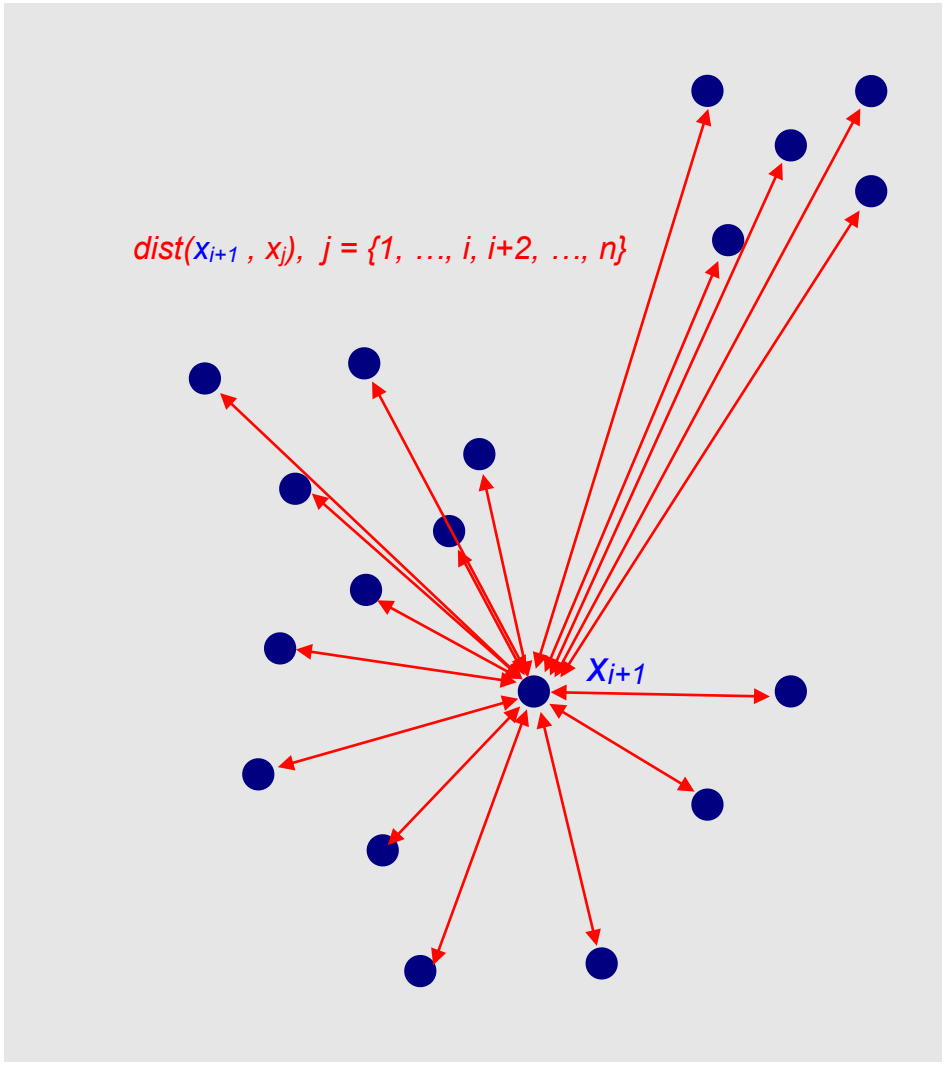


n = *20* observations
p = *2* variables (dimension)
Pas de label

Peut-on raisonnablement distinguer plusieurs sous-groupes d'individus ?

Distance entre les observations
→ $dist(x_i , x_j),\ i, j = \{1, \ldots, n\}^2$

Exemple introductif 2 : Apprentissage non-supervisé — *Distance entre les observations*

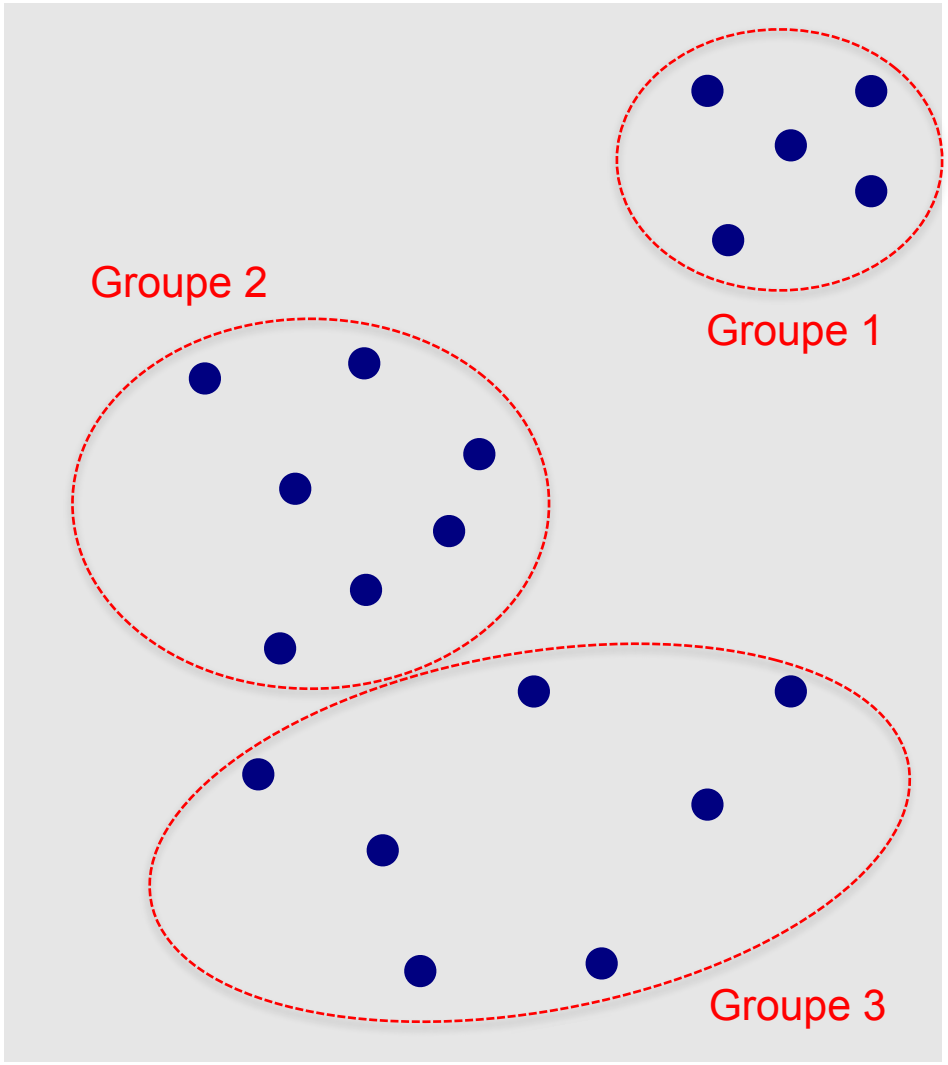


n = *20* observations
p = *2* variables (dimension)
Pas de label

Peut-on raisonnablement distinguer plusieurs sous-groupes d'individus ?

Distance entre les observations
→ $dist(x_i , x_j),\ i, j = \{1, \ldots, n\}^2$

Exemple introductif 2 : Apprentissage non-supervisé — *Apprentissage*

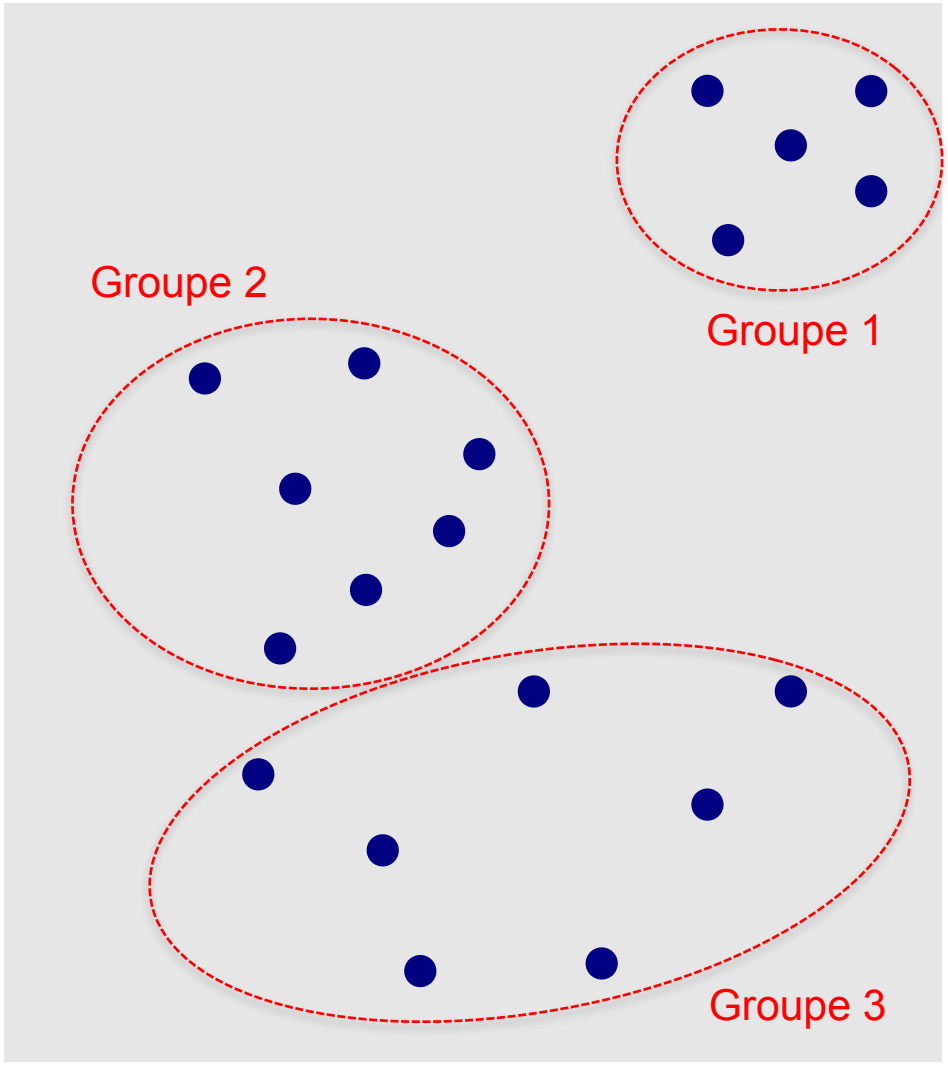


$n = 20$ observations
$p = 2$ variables (dimension)
Pas de label

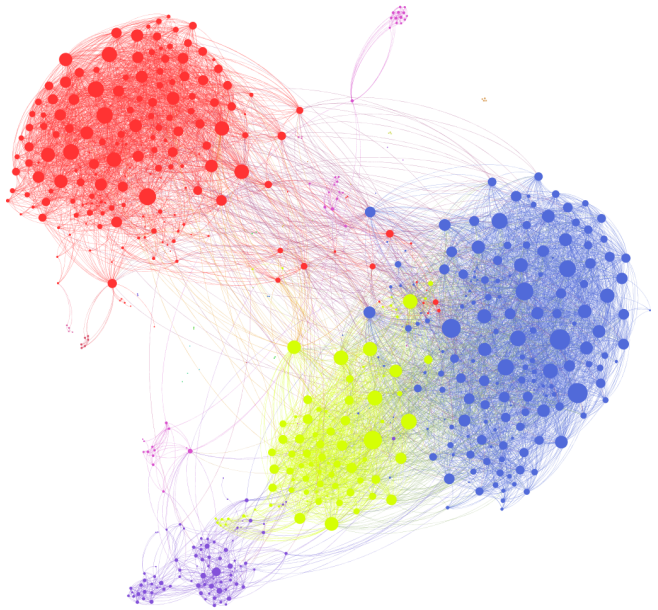
Peut-on raisonnablement distinguer plusieurs sous-groupes d'individus ?

Distance entre les observations
→ $dist(x_i, x_j),\ i, j = \{1, \ldots, n\}^2$
Energie à minimiser en fonction de labels y_i
→ $f(y_1, \ldots, y_n, \{ dist(x_i, x_j),\ i, j = \{1, \ldots, n\}^2 \})$

Exemple introductif 2 : Apprentissage non-supervisé — *Lien avec le clustering de graphe*

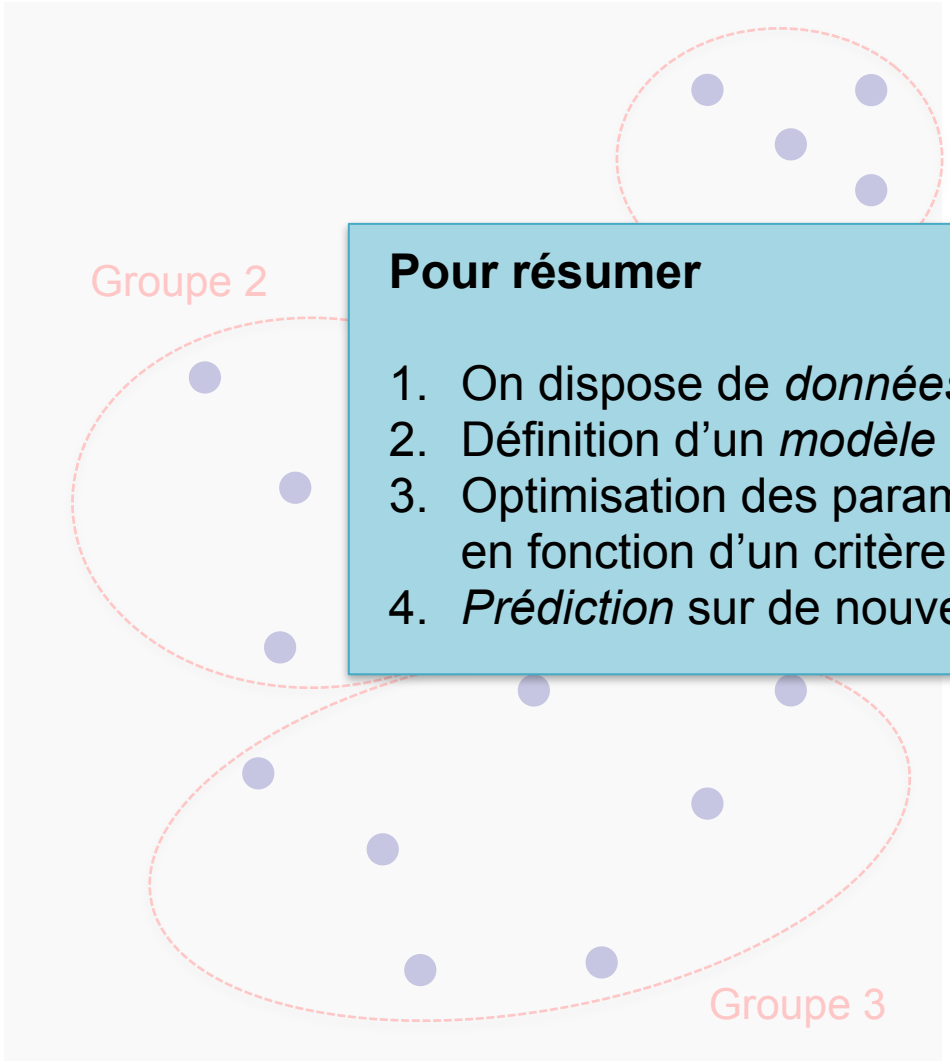


Lien fort



Graph clustering

Exemple introductif 2 : Apprentissage non-supervisé — *Apprentissage*



Pour résumer

1. On dispose de *données d'apprentissage* ***non-annotées***
2. Définition d'un *modèle* pour séparer les données
3. Optimisation des paramètres du modèle (*apprentissage*) en fonction d'un critère **d'énergie interne**
4. *Prédiction* sur de nouvelles observation

Peut-on raisonnablement distinguer plusieurs sous-groupes d'individus ?

→ Apprentissage en fonction d'une notion de distance entre les individus

Plan de la présentation

- Exemples introductifs
  - Apprentissage supervisé
  - Apprentissage non-supervisé
- Evolution des tendances en science des données
- Algorithmes classiques en apprentissage machine
  - K-means
  - Arbres de décision et Random forests
  - SVM
  - Régression logistique
- Sur-apprentissage et validation croisée
  - Sur-apprentissage
  - Validation croisée
- Grande dimension et régularisation
  - Modélisation
  - Effet de la régularisation
- Réduction de dimension par ACP
- Apprentissage machine et GPU
  - Introduction
  - Réseaux de neurones profonds (deep learning)
  - Calcul GPU en deep learning
- Conclusion

De la statistique classique à l'apprentissage machine

1) De la statistique à l'apprentissage machine

Historique[1]:

1940-70 : *Statistiques classiques*. Question associée à une hypothèse expérimentalement réfutable avec $n \approx 30$ observations et $p < 10$ variables.

1970s : Généralisation des premiers outils informatiques. L'analyse de données explore des données plus volumineuses.

1980s : Les systèmes experts sont supplantés par l'apprentissage machine et les réseaux de neurones.

1990s : *1er changement de paradigme* : Les données ne sont plus planifiées mais sont préalablement acquises : *From Data Mining to Knowledge Discovery*.

2000s : *2eme changement de paradigme* : Le nombre de variables p explose, notamment avec les données omiques où $p>>n$. La qualité de prévision devient plus importante que la réalité du modèle devenu *boîte noire*. Problématique du fléau de la dimension.

2010s : *3eme changement de paradigme* : Le nombre d'observations n explose dans le e-commerce, la géo-localisation, … . Bases de données structurées en *cloud* et moyens de calculs regroupés en *clusters (big data)*. La notion de rapidité des algorithmes devient critique.

… explicabilité et interprétabilité des décisions des algorithmes

… données complexes (small data)

[1]Cours Apprentissage Statistique de P. Besse (INSA Toulouse): http://www.math.univ-toulouse.fr/~besse/enseignement.html

1) De la statistique à l'apprentissage machine

Historique[1]:

1940-70 : *Statistiques classiques*. Question associée à une hypothèse expérimentalement réfutable avec $n \approx 30$ observations et $p < 10$ variables.

1970s : Généralisation des premiers outils informatiques. L'analyse de données explore des données plus volumineuses.

1980s : Les systèmes experts sont supplantés par l'apprentissage machine et les réseaux de neurones.

1990s : *1er changement de paradigme* : Les données ne sont plus planifiées mais sont préalablement acquises : *From Data Mining to Knowledge Discovery*.

2000s : *2eme changement de paradigme* : Le nombre de variables p explose, notamment avec les données omiques où $p>>n$. La qualité de prévision devient plus importante que la réalité du modèle devenu *boîte noire*. Problématique du fléau de la dimension.

2010s : *3eme changement de paradigme* : Le nombre d'observations n explose dans le e-commerce, la géo-localisation, … . Bases de données structurées en *cloud* et moyens de calculs regroupés en *clusters (big data)*. La notion de rapidité des algorithmes devient critique.

… explicabilité et interprétabilité des décisions des algorithmes

… données complexes (small data)

[1]Cours Apprentissage Statistique de P. Besse (INSA Toulouse): http://www.math.univ-toulouse.fr/~besse/enseignement.html

1) De la statistique à l'apprentissage machine

Historique[1]:

1940-70 : *Statistiques classiques*. Question associée à une hypothèse expérimentalement réfutable avec $n \approx 30$ observations et $p < 10$ variables.

1970s : Généralisation des premiers outils informatiques. L'analyse de données explore des données plus volumineuses.

1980s : Les systèmes experts sont supplantés par l'apprentissage machine et les réseaux de neurones.

1990s : *1er changement de paradigme* : Les données ne sont plus planifiées mais sont préalablement acquises : *From Data Mining to Knowledge Discovery*.

2000s : *2eme changement de paradigme* : Le nombre de variables p explose, notamment avec les données omiques où $p>>n$. La qualité de prévision devient plus importante que la réalité du modèle devenu *boîte noire*. Problématique du fléau de la dimension.

2010s : *3eme changement de paradigme* : Le nombre d'observations n explose dans le e-commerce, la géo-localisation, … . Bases de données structurées en *cloud* et moyens de calculs regroupés en *clusters (big data)*. La notion de rapidité des algorithmes devient critique.

… explicabilité et interprétabilité des décisions des algorithmes

… données complexes (small data)

[1]Cours Apprentissage Statistique de P. Besse (INSA Toulouse): http://www.math.univ-toulouse.fr/~besse/enseignement.html

1) De la statistique à l'apprentissage machine

Historique[1]:

1940-70 : *Statistiques classiques*. Question associée à une hypothèse expérimentalement réfutable avec $n \approx 30$ observations et $p < 10$ variables.

1970s : Généralisation des premiers outils informatiques. L'analyse de données explore des données plus volumineuses.

1980s : Les systèmes experts sont supplantés par l'apprentissage machine et les réseaux de neurones.

1990s : *1er changement de paradigme* : Les données ne sont plus planifiées mais sont préalablement acquises : *From Data Mining to Knowledge Discovery*.

2000s : *2eme changement de paradigme* : Le nombre de variables p explose, notamment avec les données omiques où $p>>n$. La qualité de prévision devient plus importante que la réalité du modèle devenu *boîte noire*. Problématique du fléau de la dimension.

2010s : *3eme changement de paradigme* : Le nombre d'observations n explose dans le e-commerce, la géo-localisation, … . Bases de données structurées en *cloud* et moyens de calculs regroupés en *clusters (big data)*. La notion de rapidité des algorithmes devient critique.

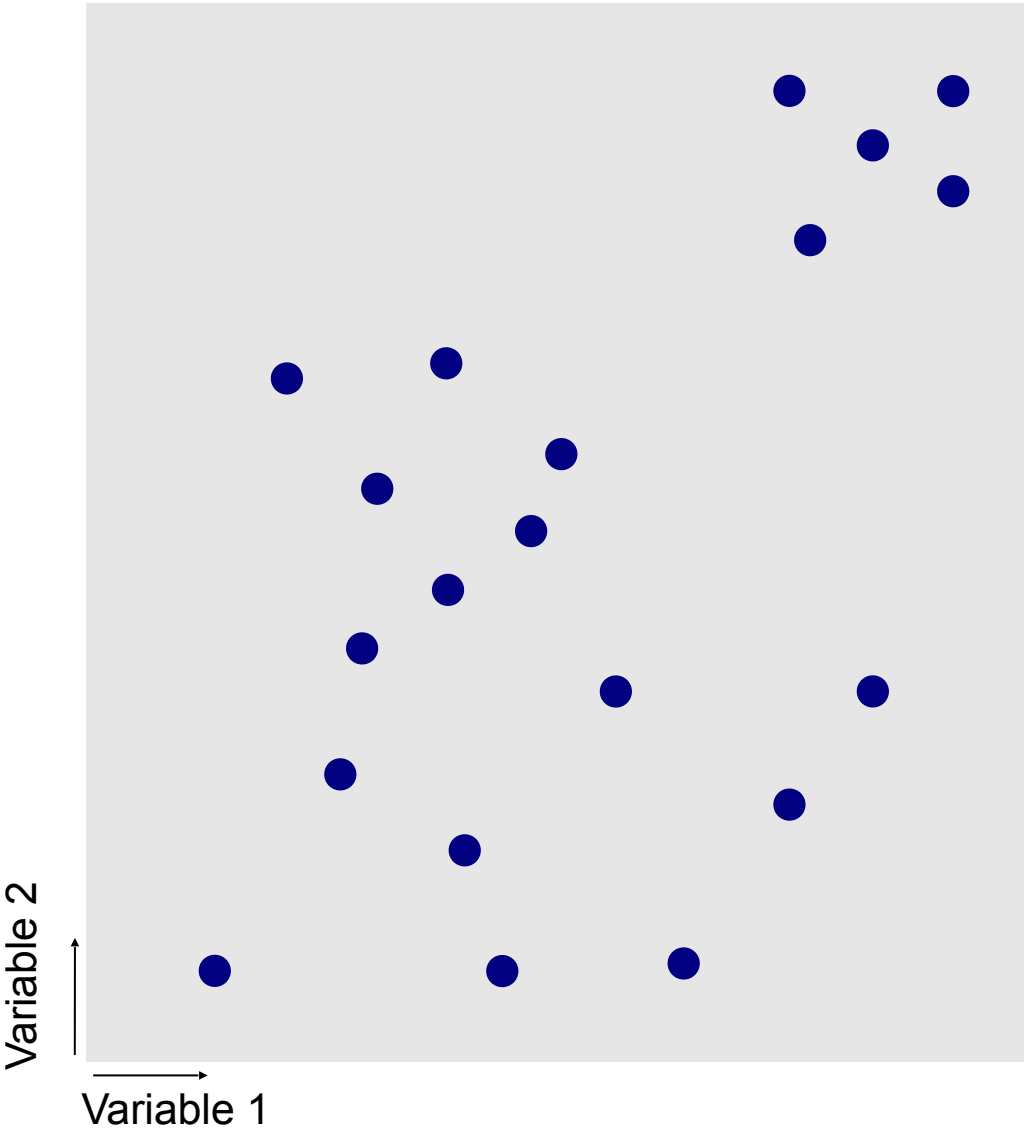
… explicabilité et interprétabilité des décisions des algorithmes

… données complexes (small data)

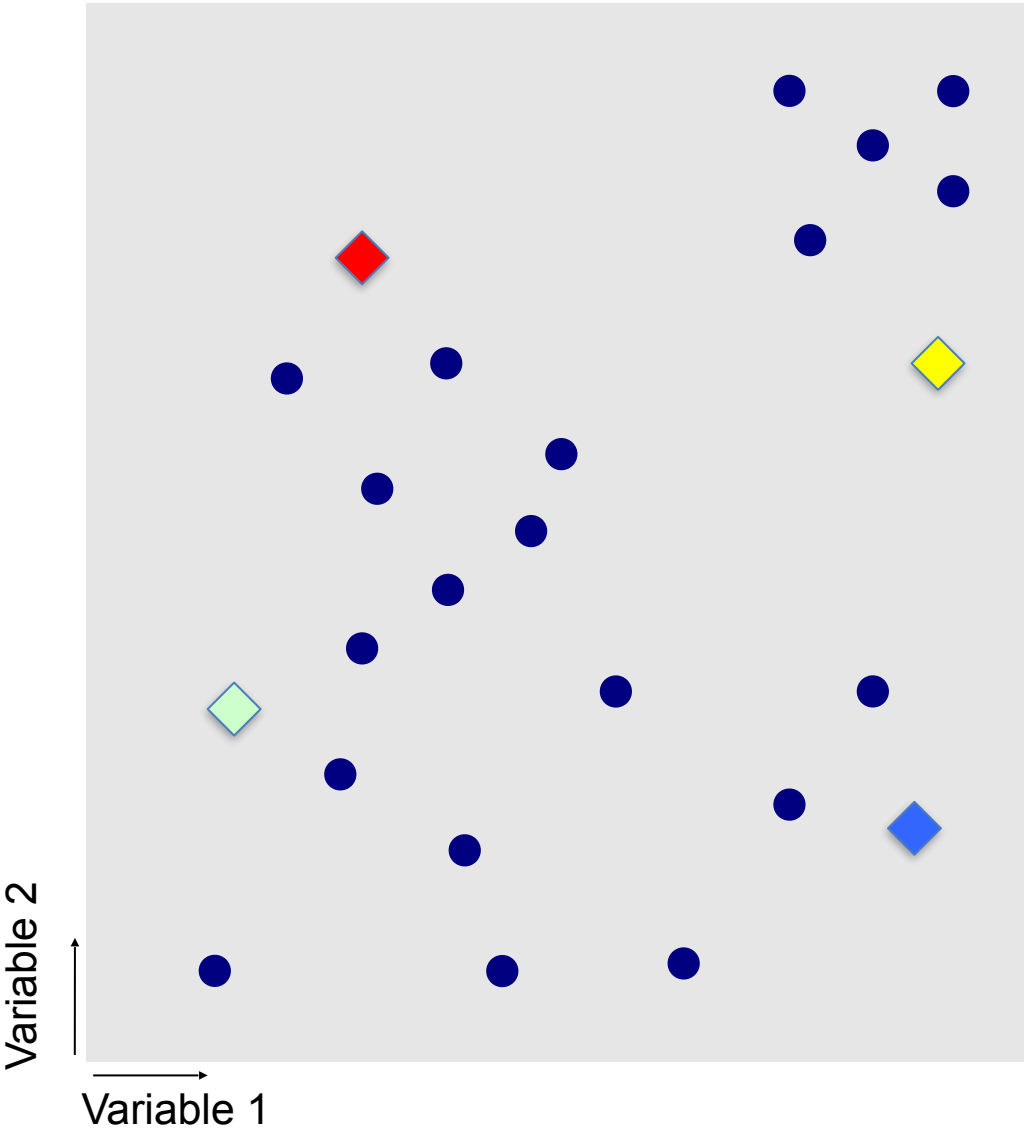
[1]Cours Apprentissage Statistique de P. Besse (INSA Toulouse): http://www.math.univ-toulouse.fr/~besse/enseignement.html

Algorithmes classiques en machine learning

Algorithme des K-means

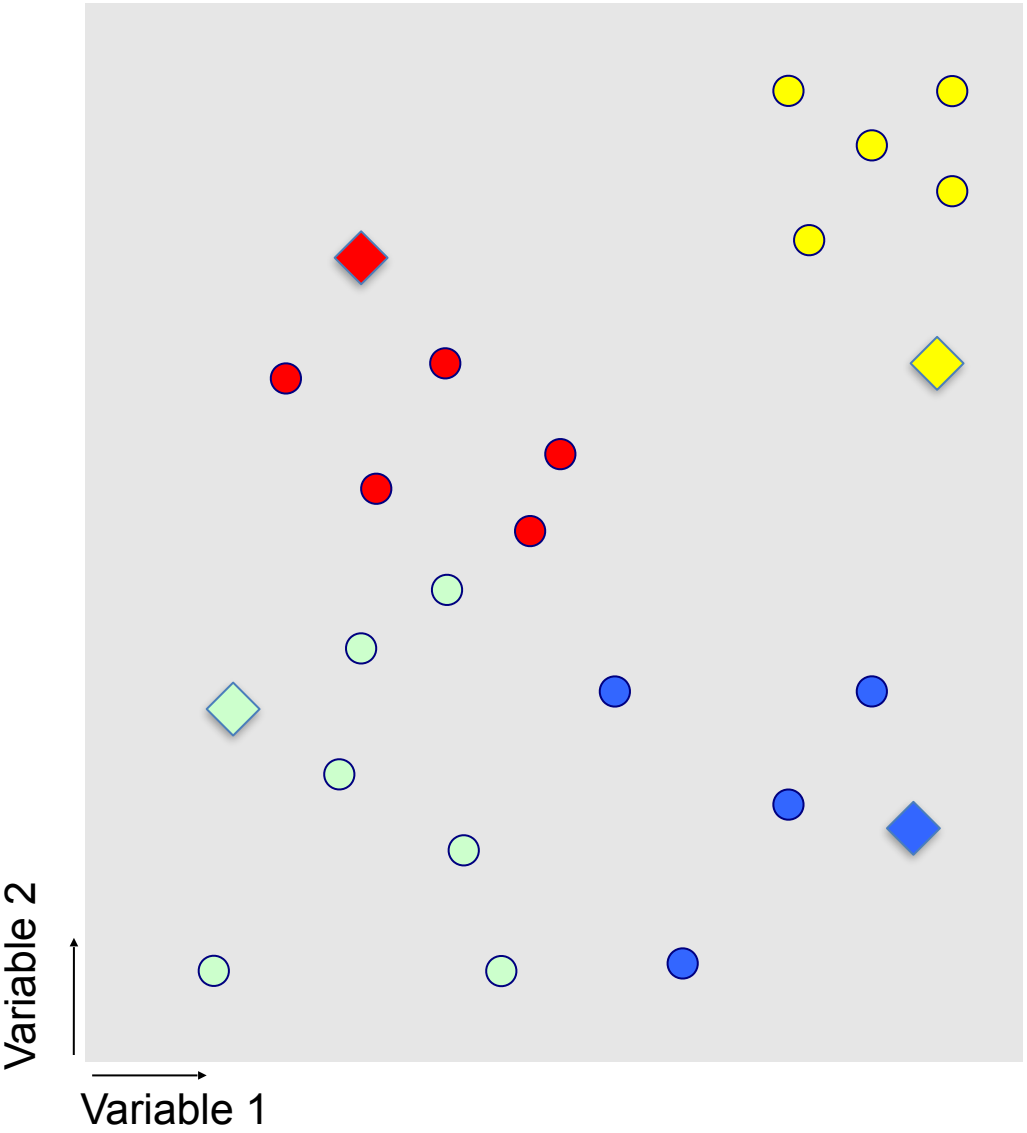


Algorithme des K-means



On tire N graines au hasard
(pour l'exemple N=4)

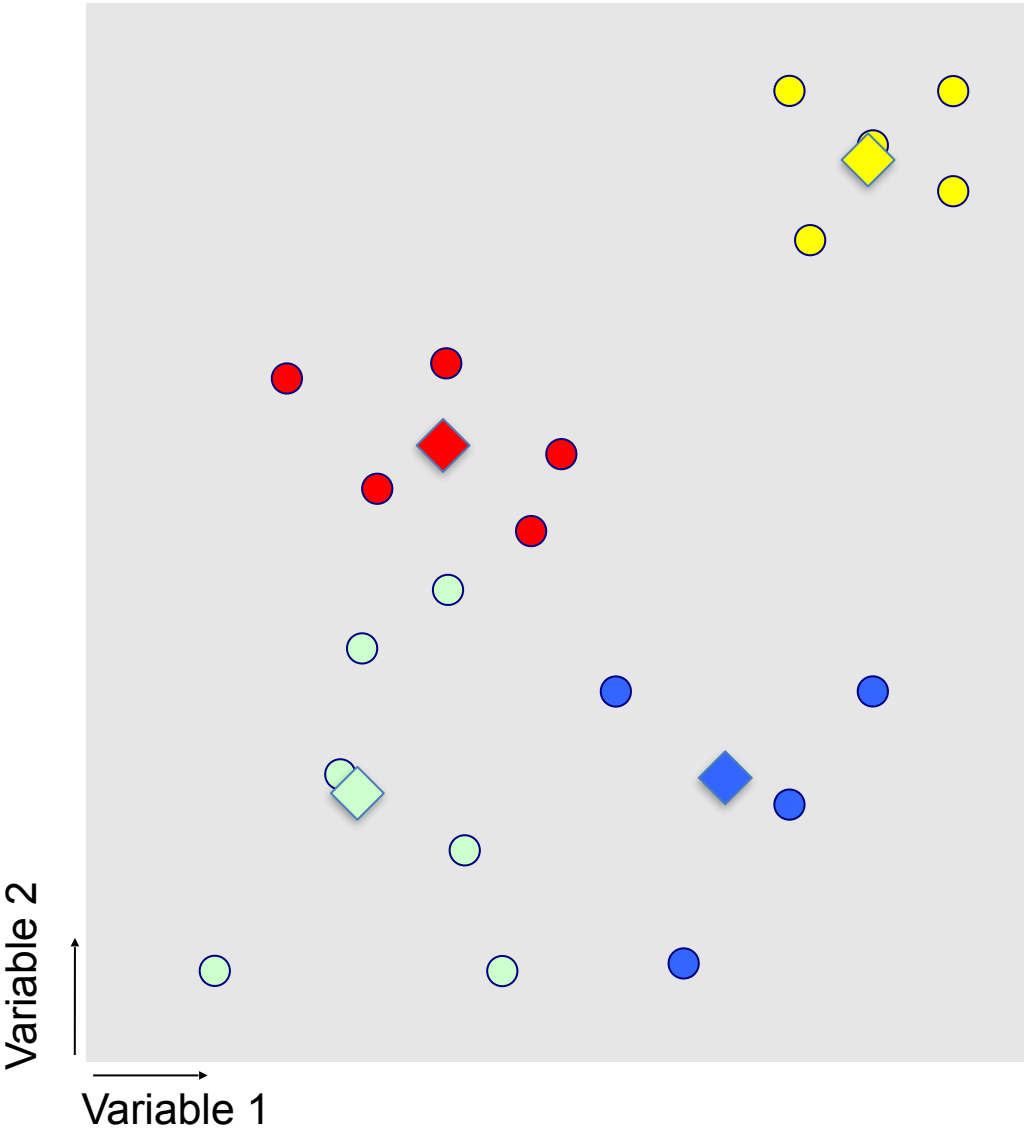
Algorithme des K-means



Pour chaque observation, on cherche la graine la plus proche.

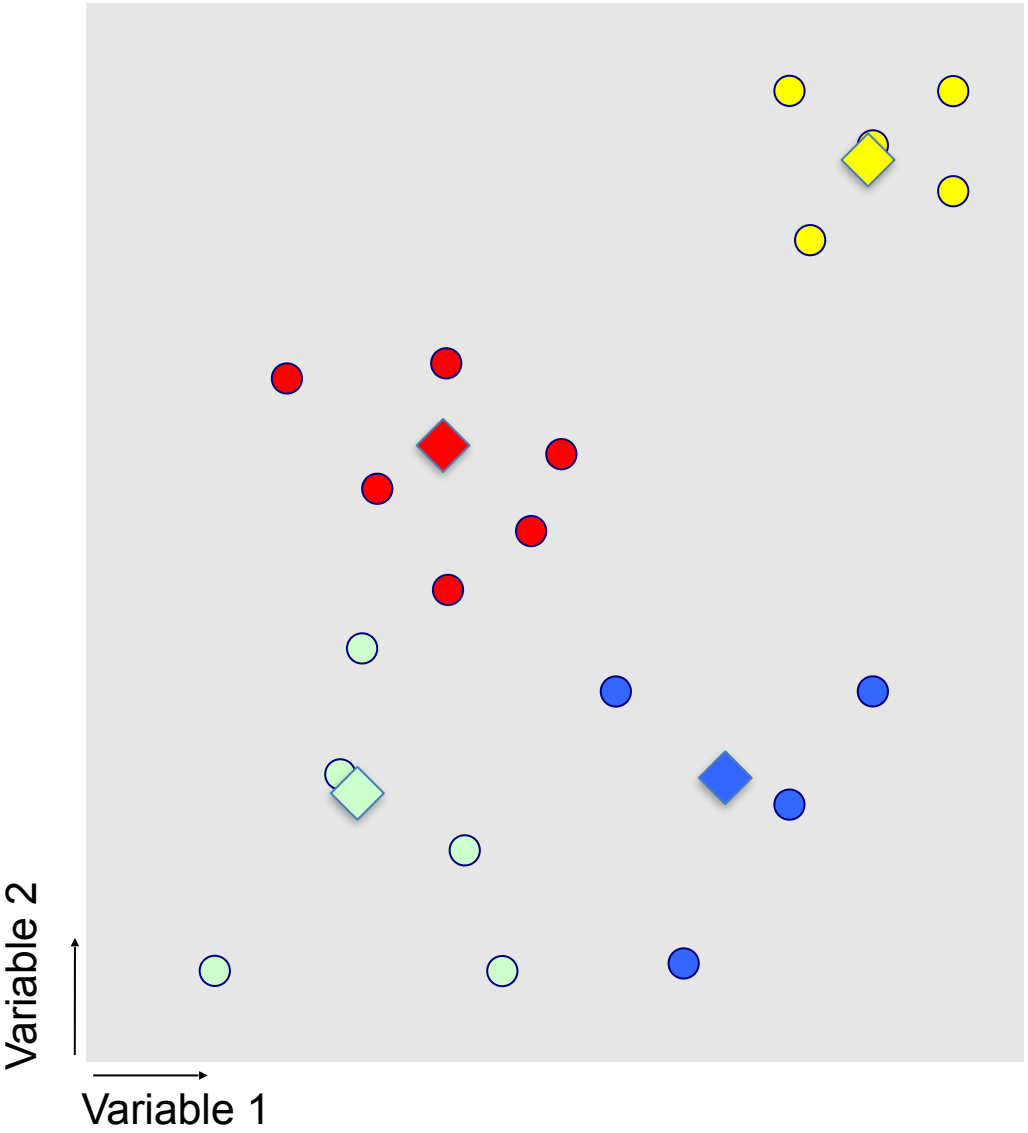
Remarque : Des distances Euclidiennes sont utilisées

Algorithme des K-means



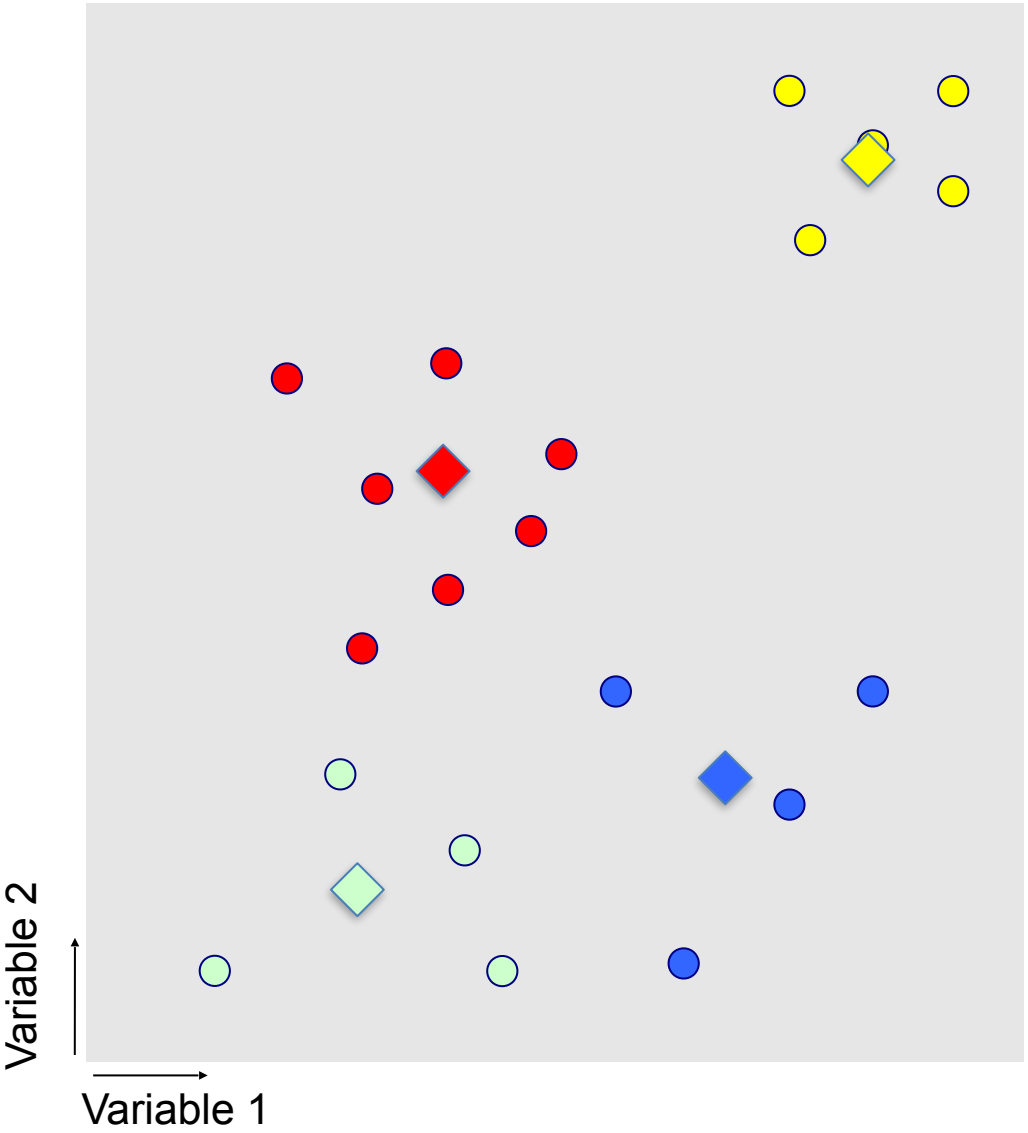
On centre les graines…

Algorithme des K-means



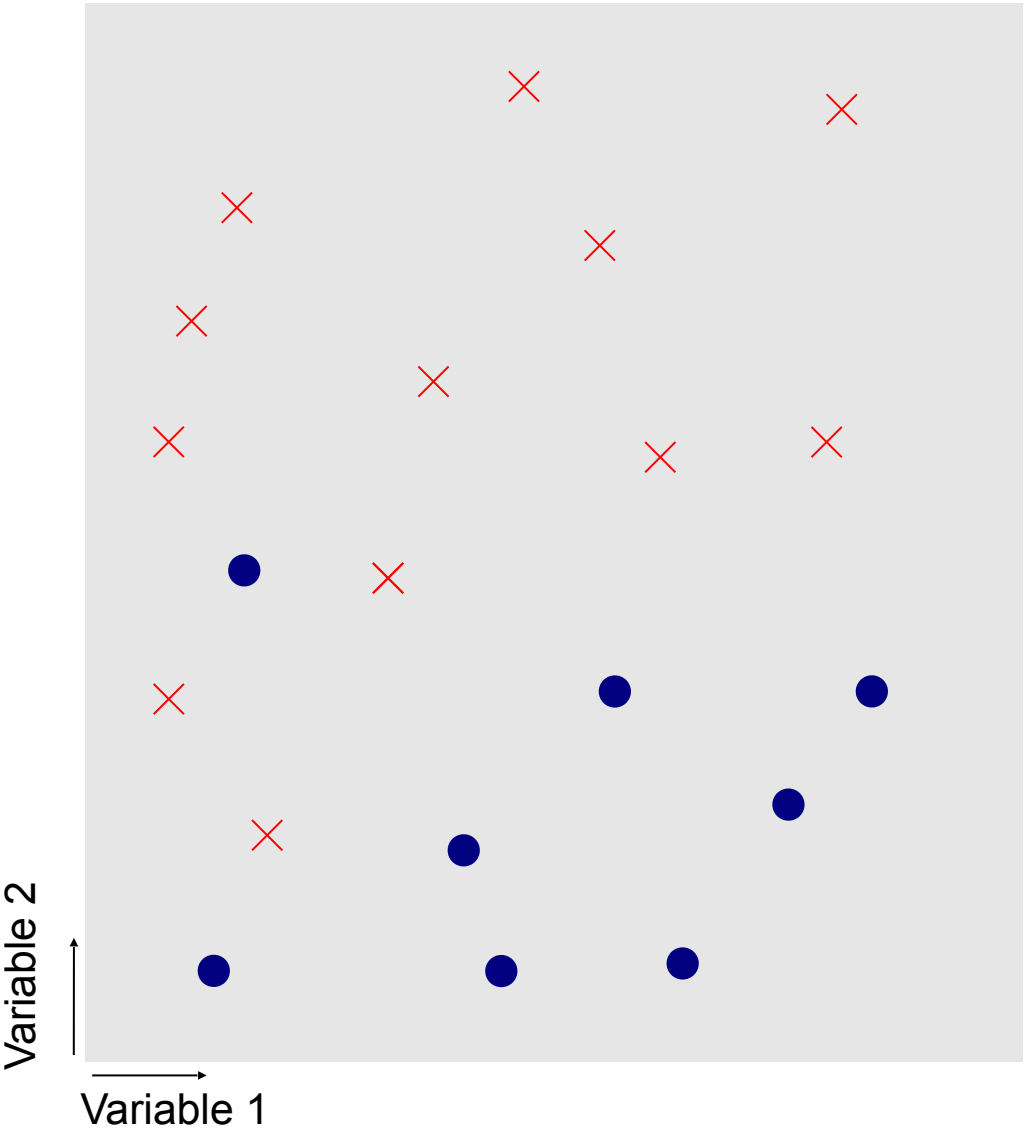
… pour chaque observation, on cherche à nouveau la graine la plus proche …

Algorithme des K-means



… et on recommence jusqu'a convergence.

Arbres de décision

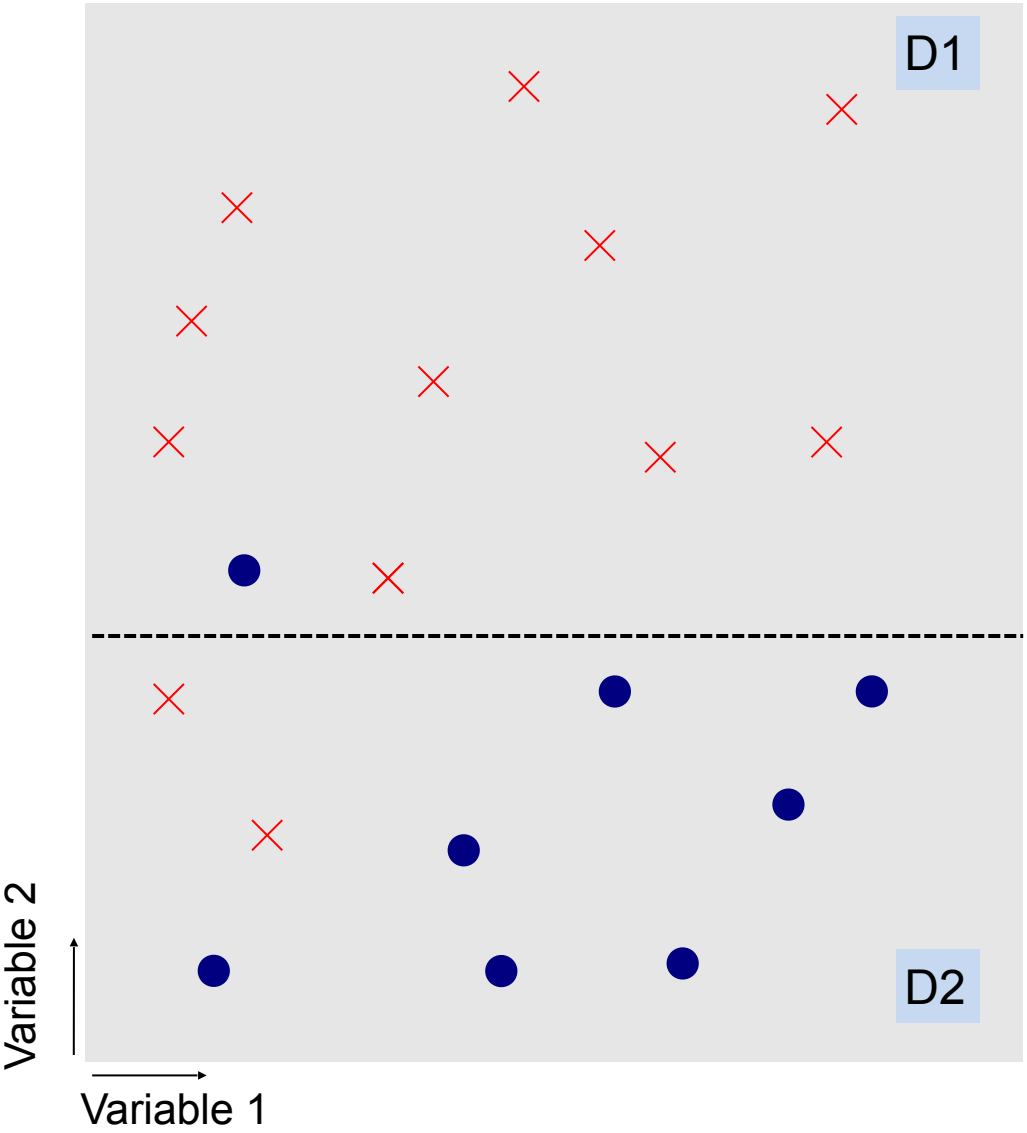


$\boldsymbol{x_1}$, $\boldsymbol{x_2}$, … , $\boldsymbol{x_N}$ les observations
(ici : x_i est la coordonnée du point en 2D)

y_1 , y_2 , … , y_N les labels
(ici : × → 1 et ● → -1)

On coupe le domaine en *sous domaines* pour minimiser la variance dans chaque sous domaine (CART).

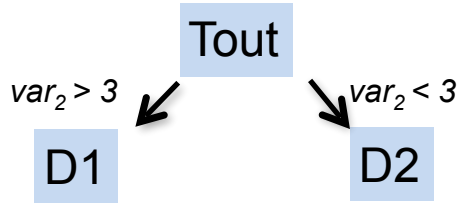
Arbres de décision



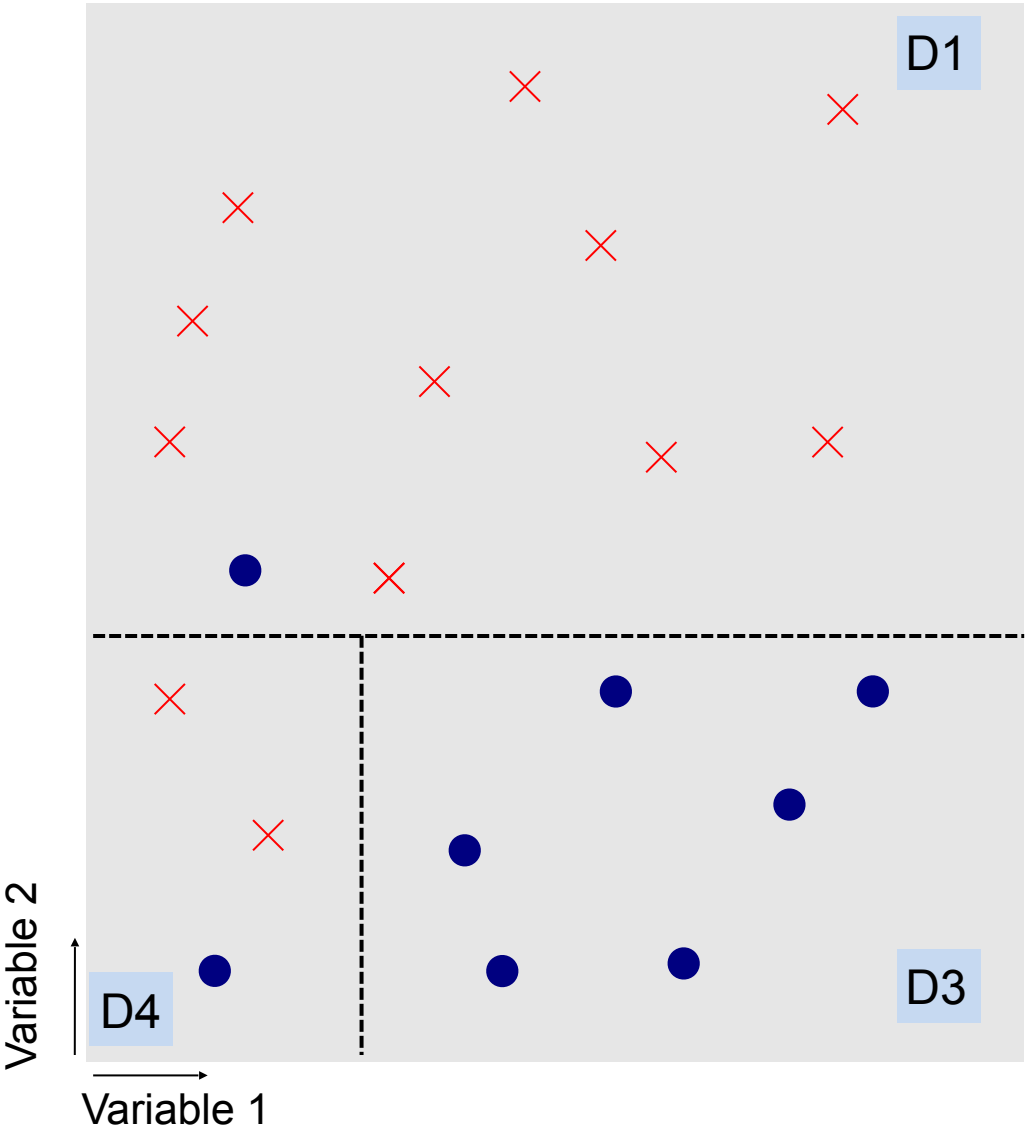
$\boldsymbol{x_1}$, $\boldsymbol{x_2}$, … , $\boldsymbol{x_N}$ les observations
(ici : x_i est la coordonnée du point en 2D)

y_1 , y_2 , … , y_N les labels
(ici : × → 1 et ● → -1)

On coupe le domaine en *sous domaines* pour minimiser la variance dans chaque sous domaine (CART).



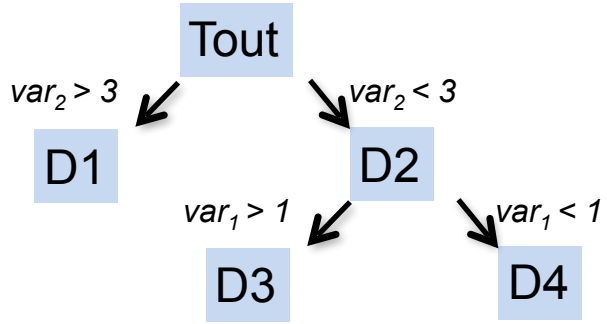
Arbres de décision



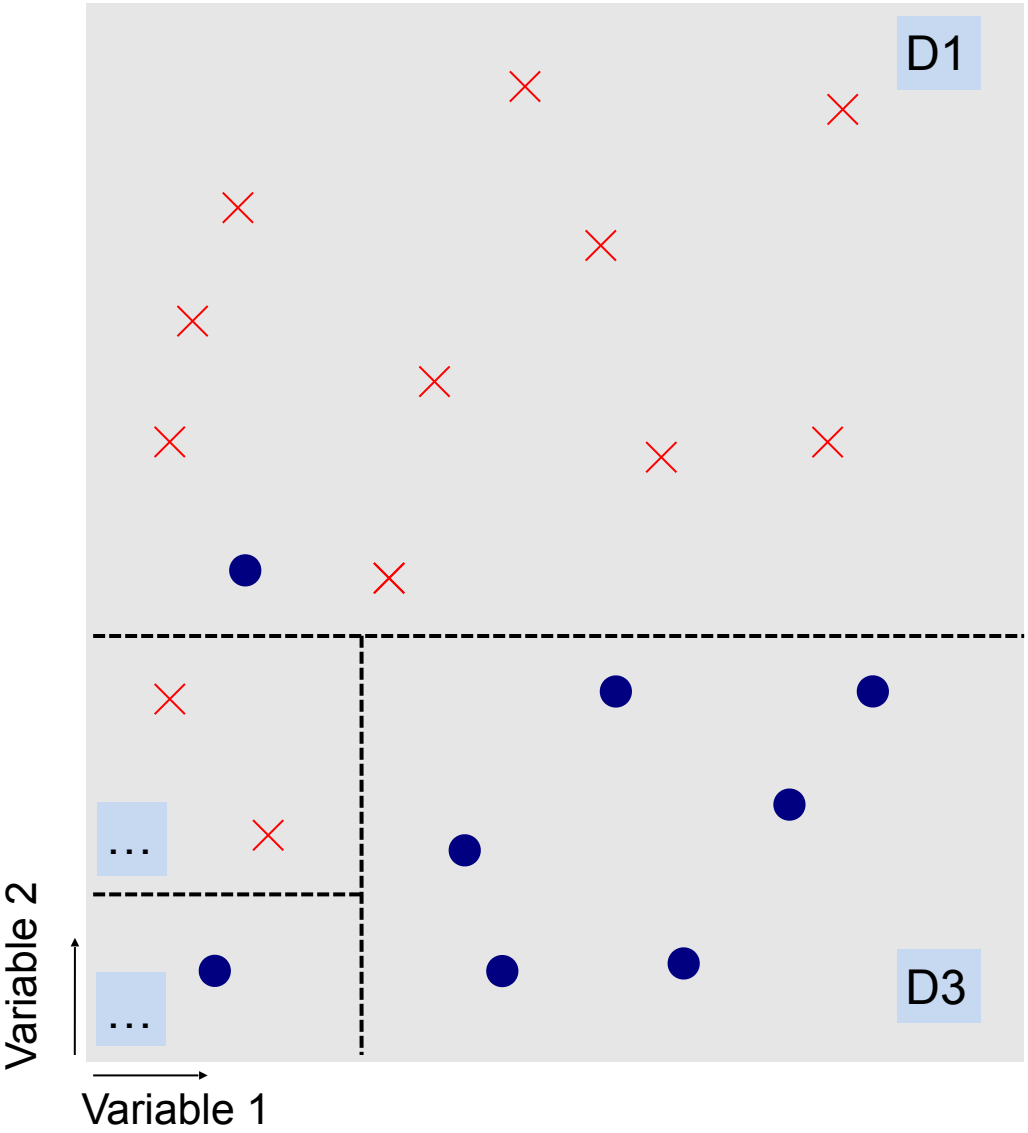
$\boldsymbol{x_1}$, $\boldsymbol{x_2}$, … , $\boldsymbol{x_N}$ les observations
(ici : x_i est la coordonnée du point en 2D)

y_1 , y_2 , … , y_N les labels
(ici : × → 1 et ● → -1)

On coupe le domaine en *sous domaines* pour minimiser la variance dans chaque sous domaine (CART).



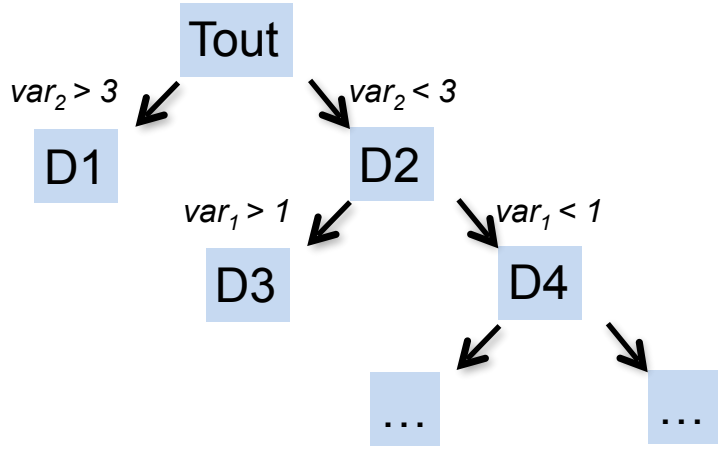
Arbres de décision



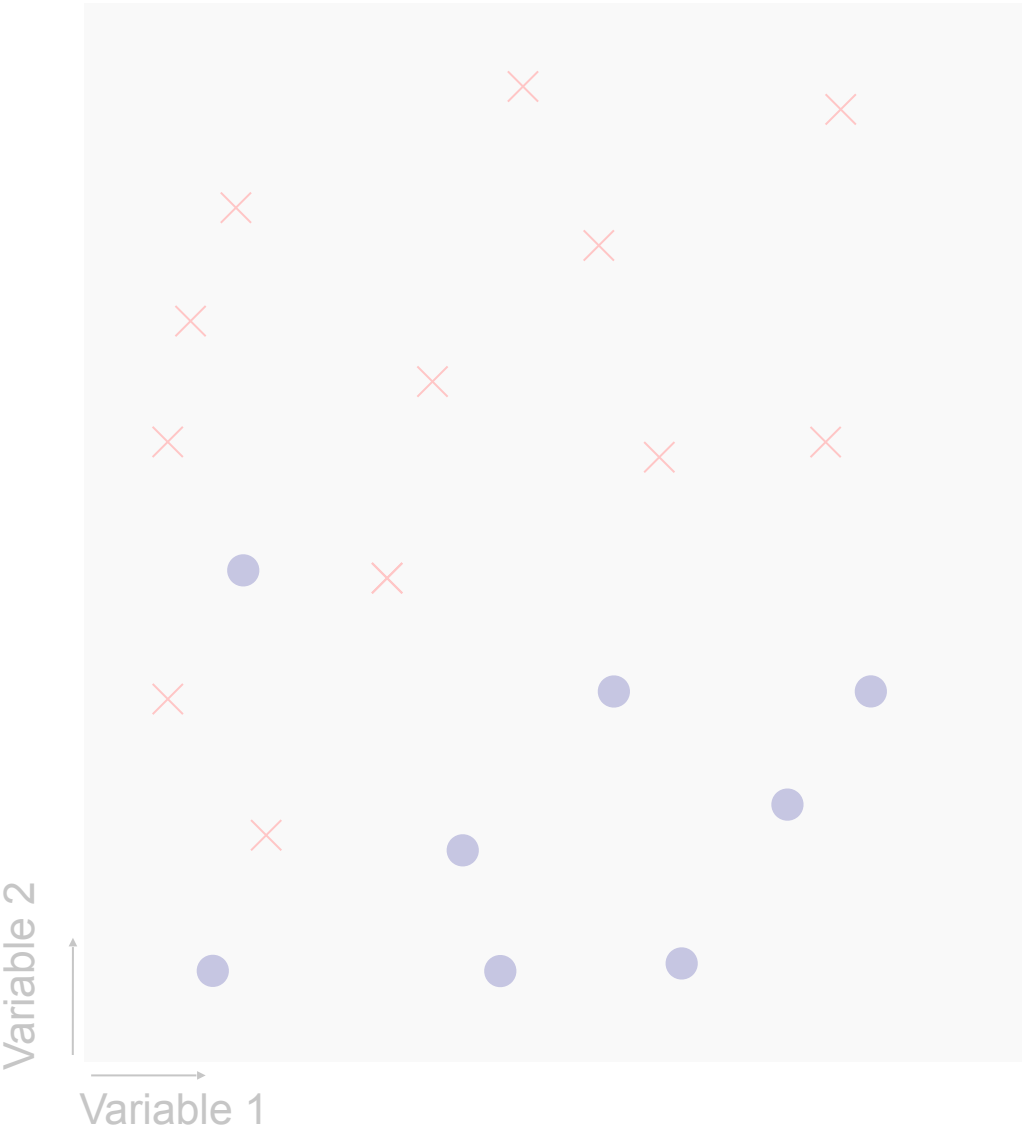
$\boldsymbol{x_1}$, $\boldsymbol{x_2}$, … , $\boldsymbol{x_N}$ les observations
(ici : x_i est la coordonnée du point en 2D)

y_1 , y_2 , … , y_N les labels
(ici : × → 1 et ● → -1)

On coupe le domaine en *sous domaines* pour minimiser la variance dans chaque sous domaine (CART).



Random forest



$x_1, x_2, \ldots, x_N$ les observations
(ici : x_i est la coordonnée du point en 2D)

$y_1, y_2, \ldots, y_N$ les labels
(ici : × ➔ 1 et ● ➔ -1)

Contexte de grande dimension : $p >> 1$ et pas $p = 2$ comme ici.

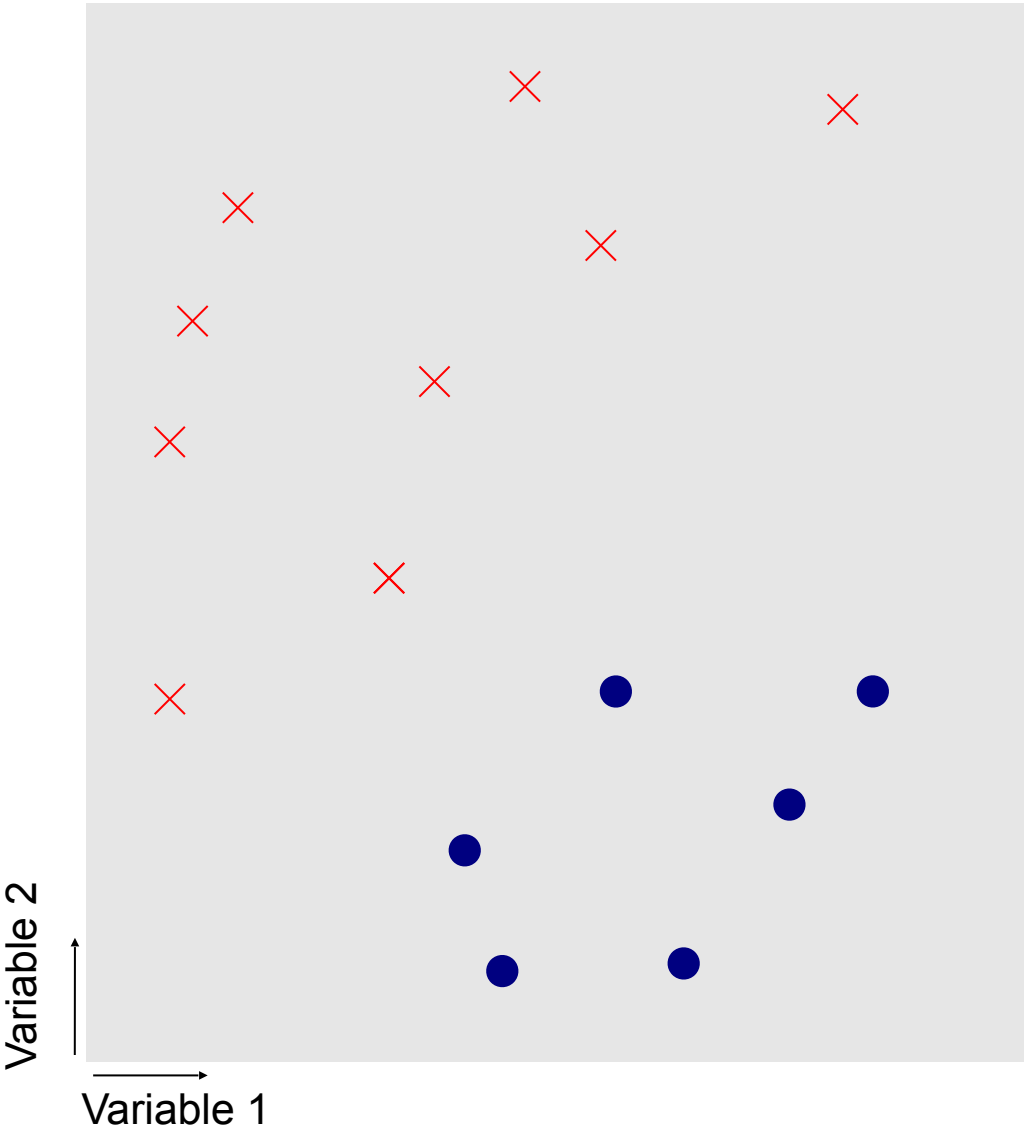
Apprentissage :

- Plusieurs arbres sont définis de manière indépendante.
- Les dimensions de coupes sont tirées au hasard.

Prédiction :

- Le label prédit en un point donné est celui prédit par la majorité des arbres (bagging).

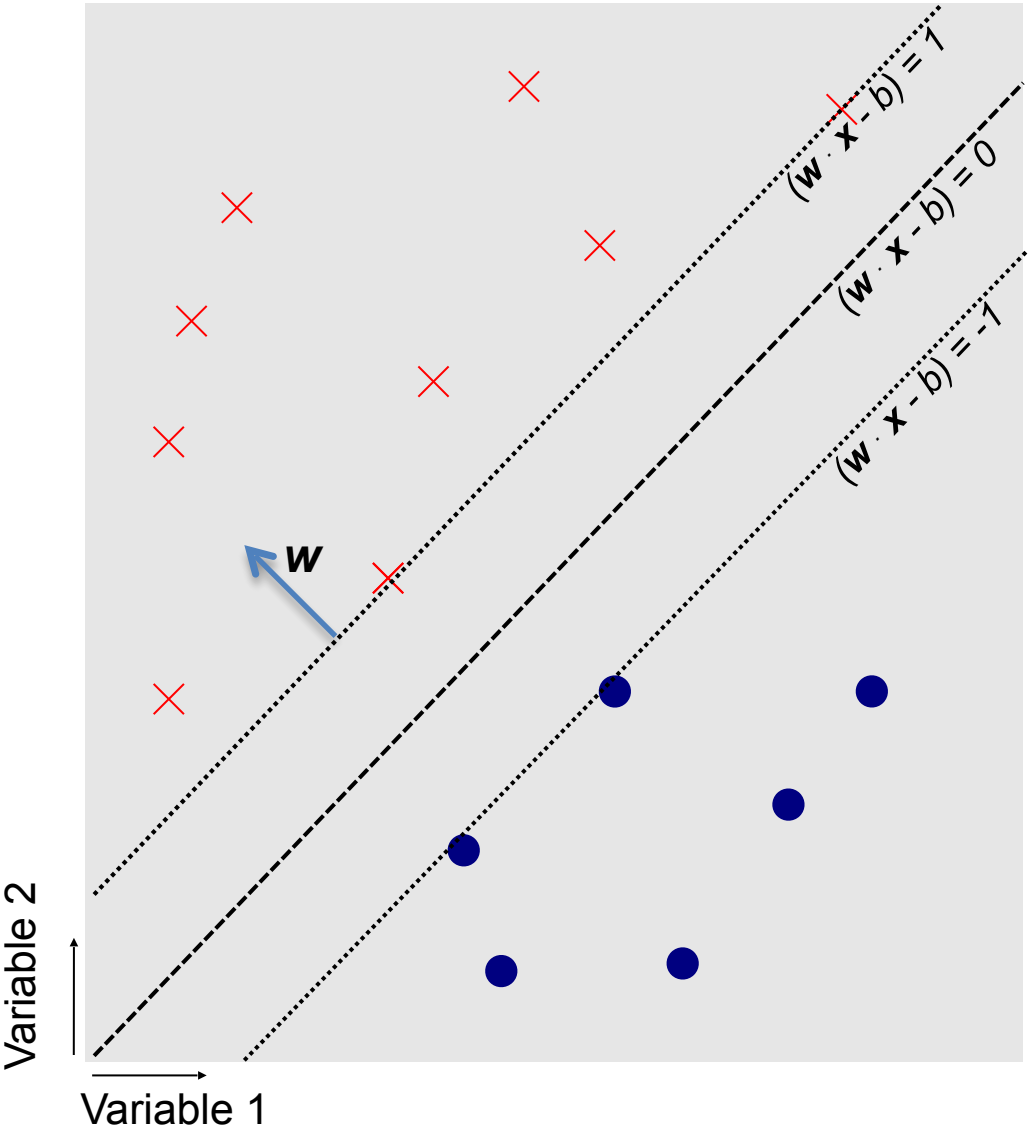
Les Support Vector Machine (SVM) — Principe



$\boldsymbol{x_1}$, $\boldsymbol{x_2}$, … , $\boldsymbol{x_N}$ les observations
(ici : x_i est la coordonnée du point en 2D)

y_1 , y_2 , … , y_N les labels
(ici : × ➔ 1 et ● ➔ -1)

Les Support Vector Machine (SVM) — Principe



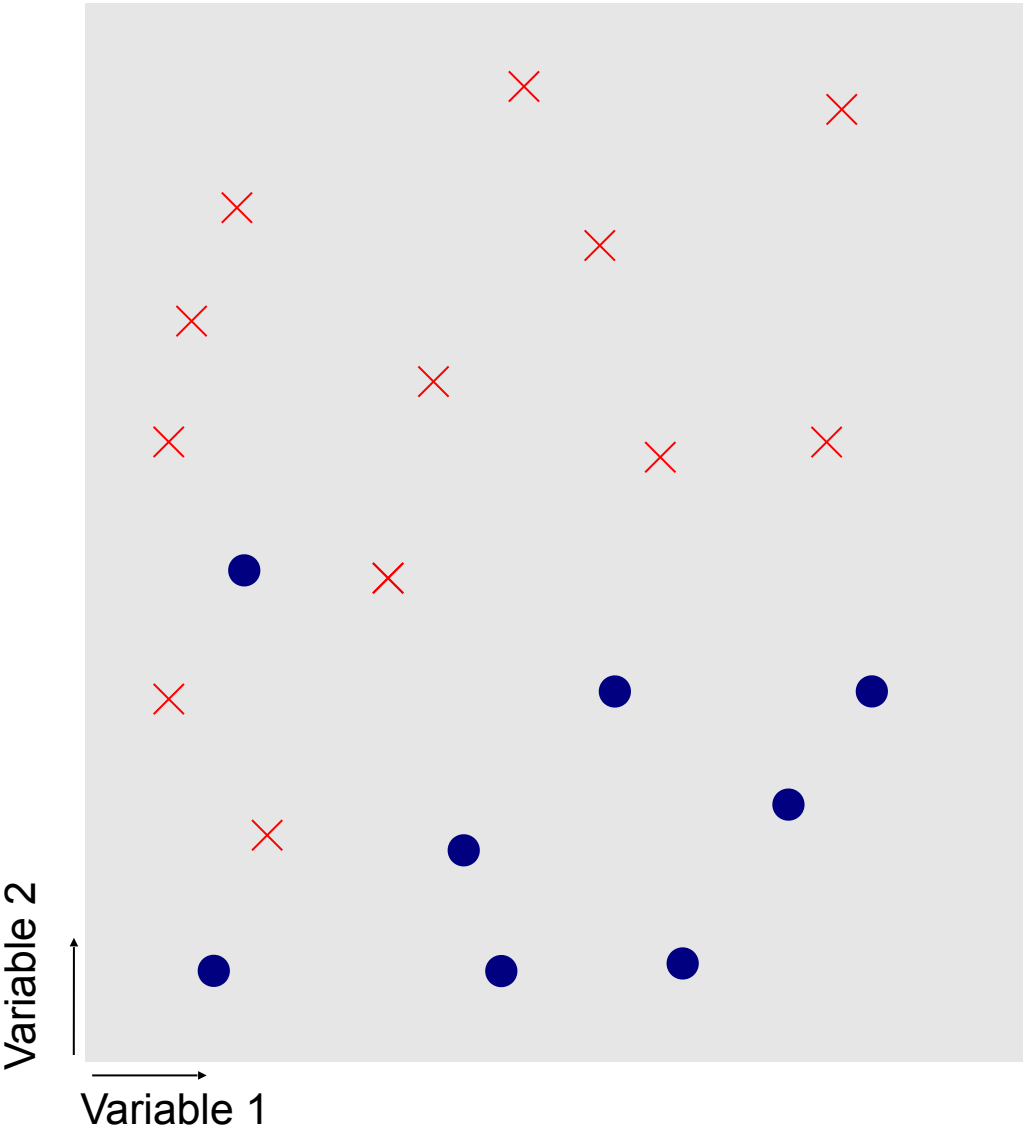
$\boldsymbol{x_1}$, $\boldsymbol{x_2}$, … , $\boldsymbol{x_N}$ les observations
(ici : x_i est la coordonnée du point en 2D)

y_1 , y_2 , … , y_N les labels
(ici : × → 1 et ● → -1)

On optimise $\boldsymbol{w}$ et b tel que :

$y_i\ (\boldsymbol{w} \cdot \boldsymbol{x_i} - b) \geq 1$ pour tout $1 \leq i \leq n$

Les Support Vector Machine (SVM) — Formulation

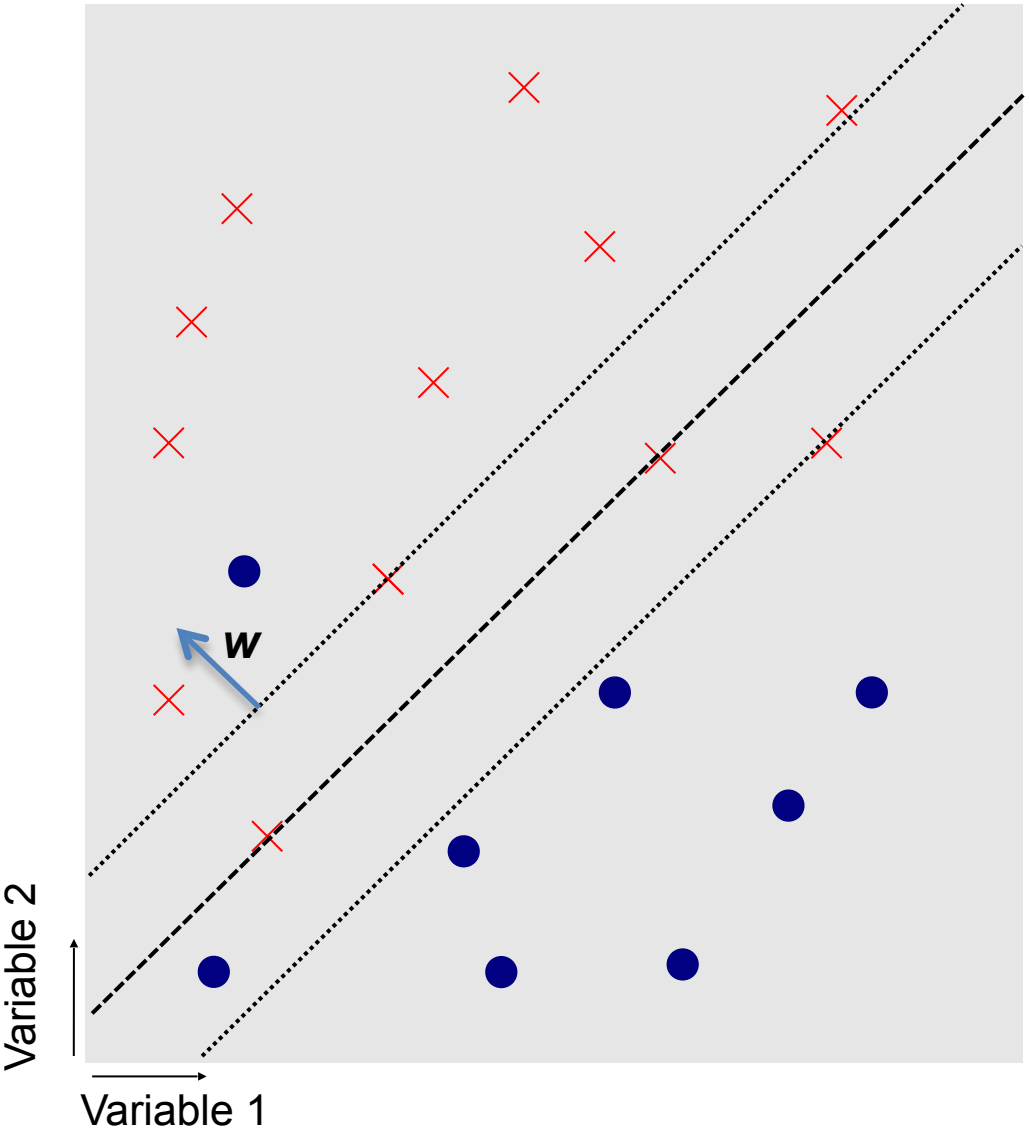


$\boldsymbol{x_1}$, $\boldsymbol{x_2}$, … , $\boldsymbol{x_N}$ les observations
(ici : x_i est la coordonnée du point en 2D)

y_1 , y_2 , … , y_N les labels
(ici : × → 1 et ● → -1)

Que faire si il est impossible de séparer tous les points?

Les Support Vector Machine (SVM) — Formulation



$\boldsymbol{x_1}$, $\boldsymbol{x_2}$, … , $\boldsymbol{x_N}$ les observations
(ici : x_i est la coordonnée du point en 2D)

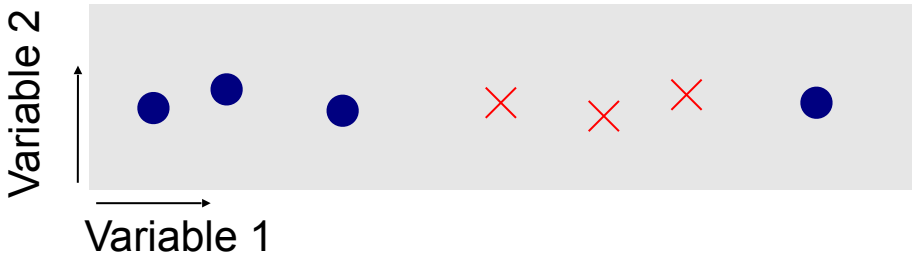
y_1 , y_2 , … , y_N les labels
(ici : × ➔ 1 et ● ➔ -1)

Que faire si il est impossible de séparer tous les points?

On cherche $\boldsymbol{w}$ et b qui minimisent :

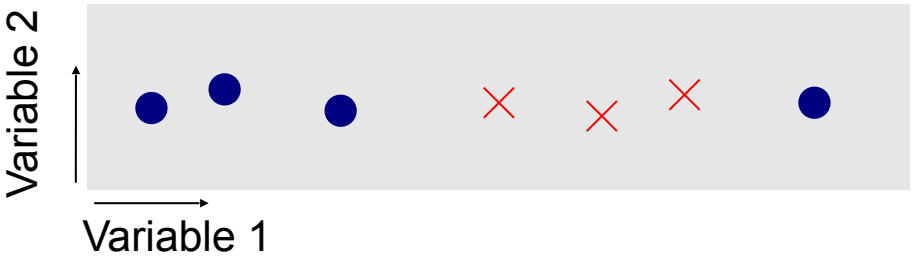
$$\left[n^{-1} \sum_{i=1}^{n} max(0 , \underbrace{1 - y_i (\boldsymbol{w} \cdot \boldsymbol{x_i} - b)}_{> 0 \text{ si } \boldsymbol{x_i} \text{ est mal classé}}) \right] + \lambda \, || \boldsymbol{w} ||_2^2$$

Les Support Vector Machine (SVM) — Méthodes à noyaux



Que faire dans ce cas là ?

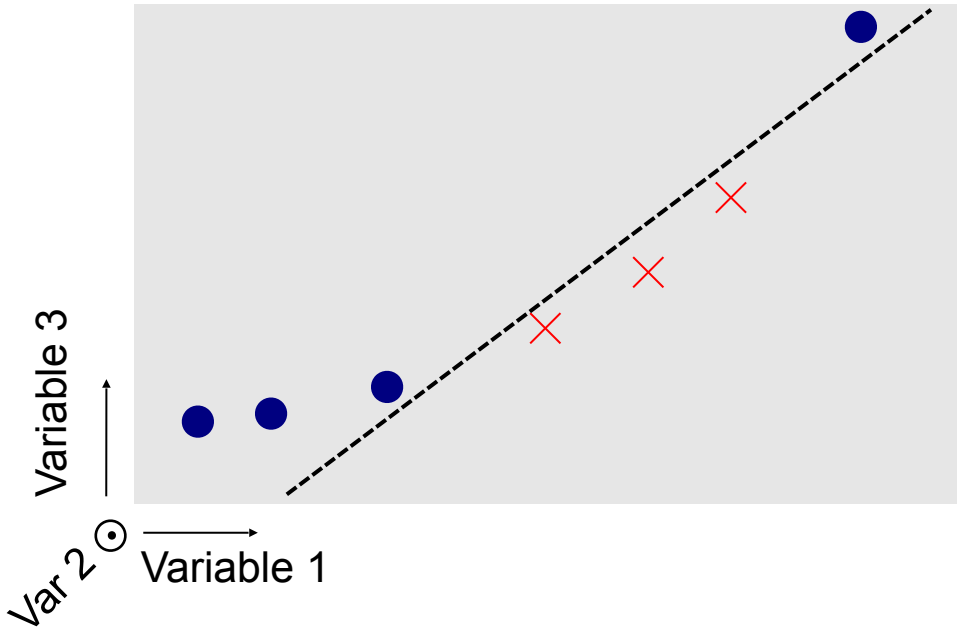
Les Support Vector Machine (SVM) — Méthodes à noyaux



Que faire dans ce cas là ?

On note $x_i = (x_i^1, x_i^2)$ une observation

On va séparer les $\Phi(x_i) = \left(x_i^1, x_i^2, (x_i^1)^2\right)$ plutôt que les x_i



Régression logistique

Pour chaque observation $i \in \{1,\ldots,n\}$:

- Variable explicative : $x_i = (x_i^1, x_i^2, \ldots, x_i^p)$ avec $p >> 1$
- Variable de réponse : $y_i \in \{-1,1\}$

Estimation de $\beta = (\beta_0, \beta_1, \ldots, \beta_n)$ tel que :

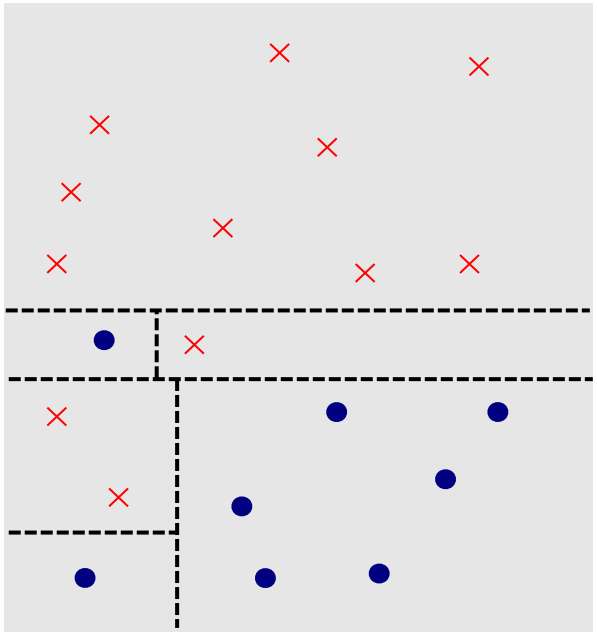
- $\beta_0 + \sum_{j=1}^{p} \beta_j x_i^j$ est positif si $y_i = 1$
- $\beta_0 + \sum_{j=1}^{p} \beta_j x_i^j$ est négatif si $y_i = -1$

Remarques :

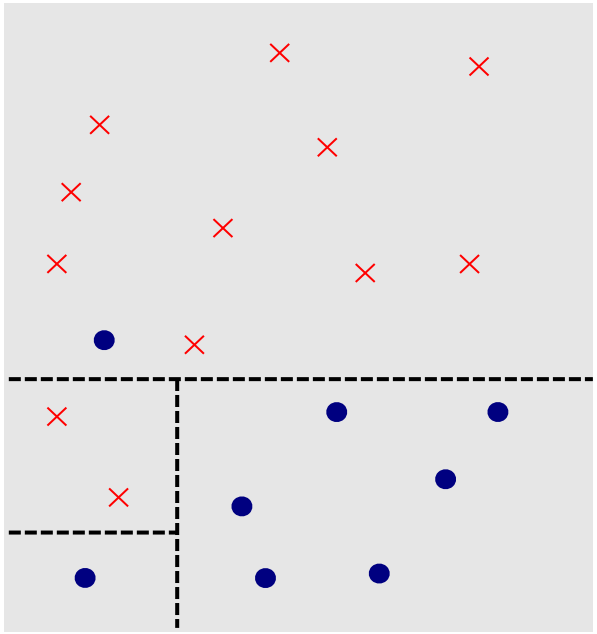
- Estimation au sens de la Log-vraisemblance en fonction d'un modèle logistique
- Contraintes possibles (et courantes) sur β en particulier quand $p > n$
- Passe à l'échelle quand *n* est très grand

Sur-apprentissage et validation croisée

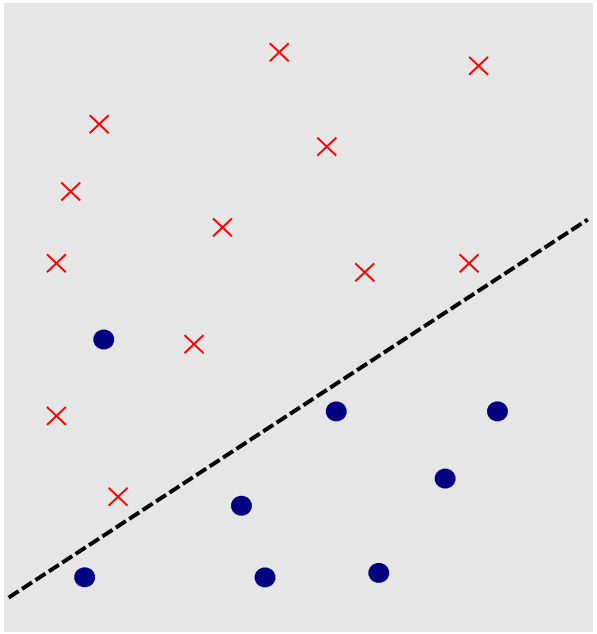
Sur-apprentissage



Arbre de décision complet

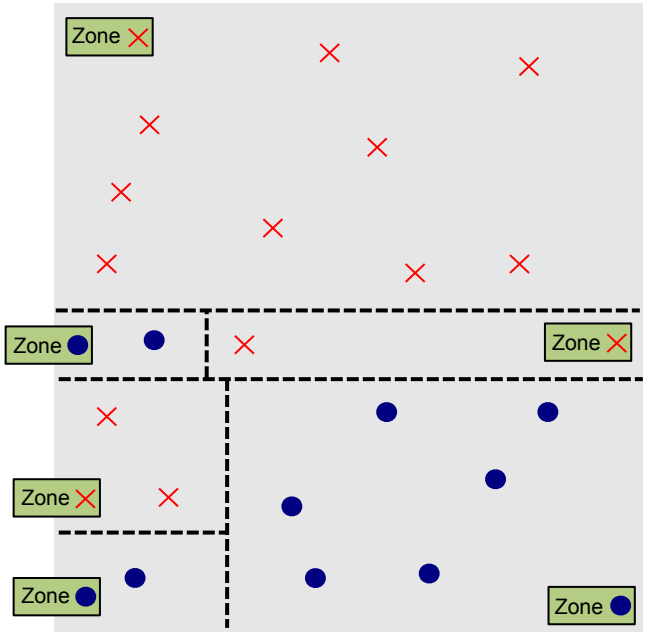


Arbre de décision incomplet

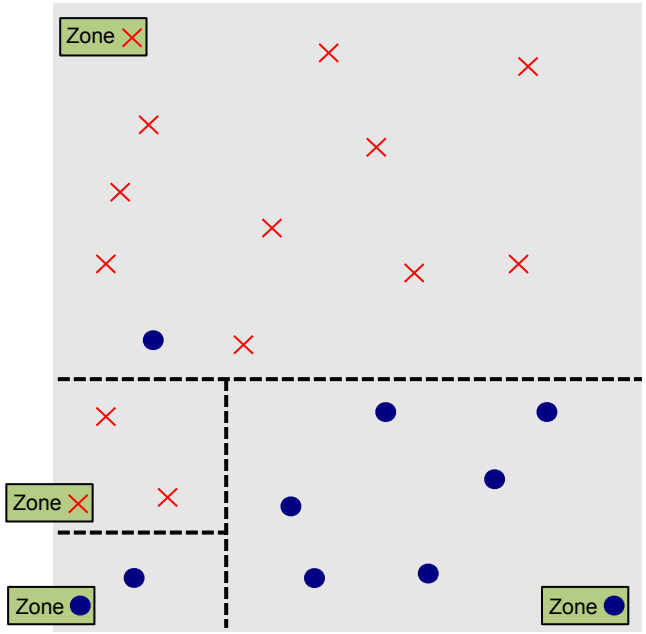


SVM linéaire

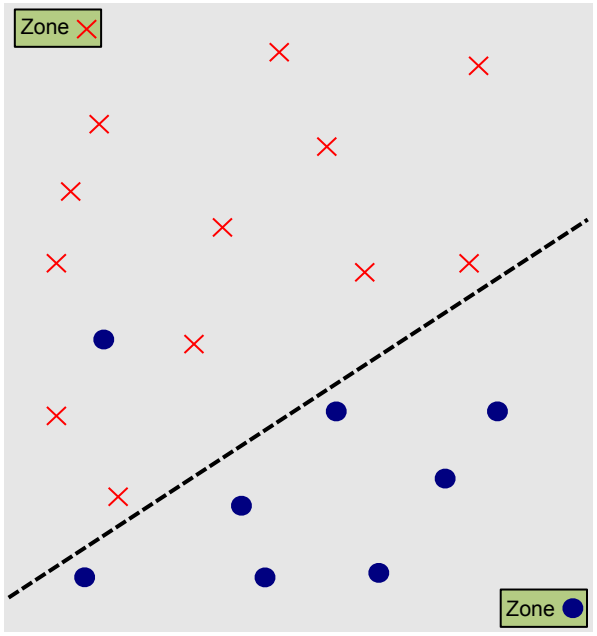
Sur-apprentissage



Arbre de décision complet



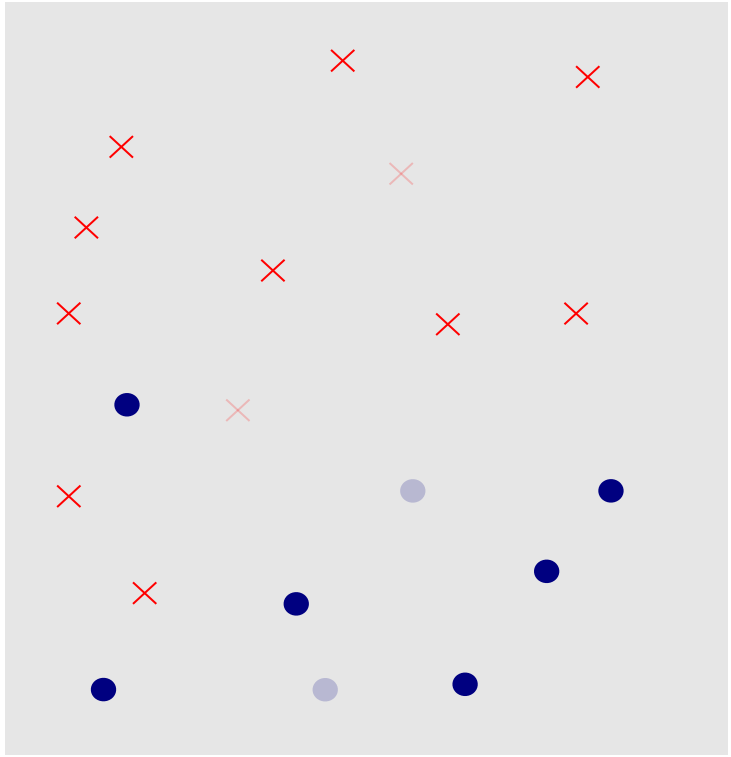
Arbre de décision incomplet



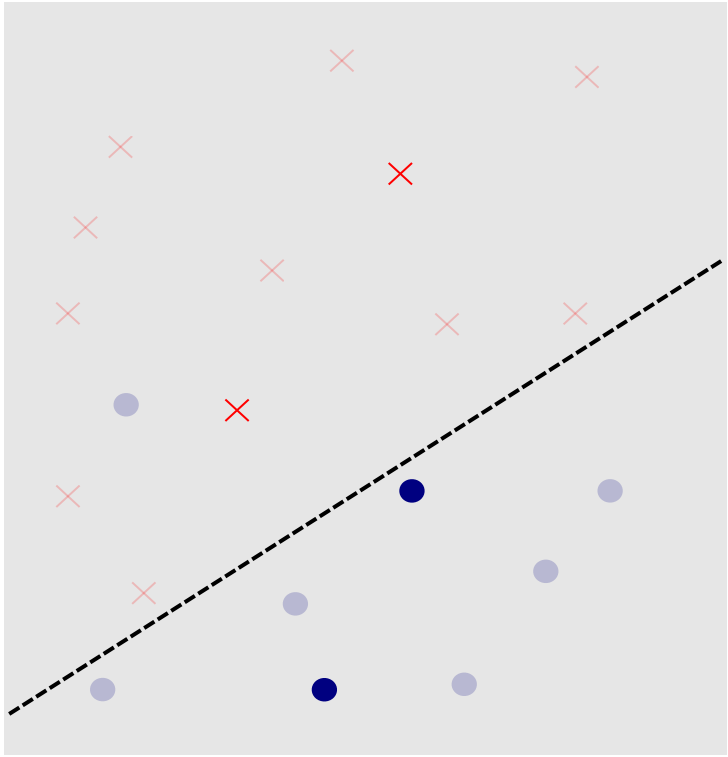
SVM linéaire

A quelle stratégie feriez-vous le plus confiance pour prédire le label d'une nouvelle observation ?

Séparation des données d'apprentissage et données test

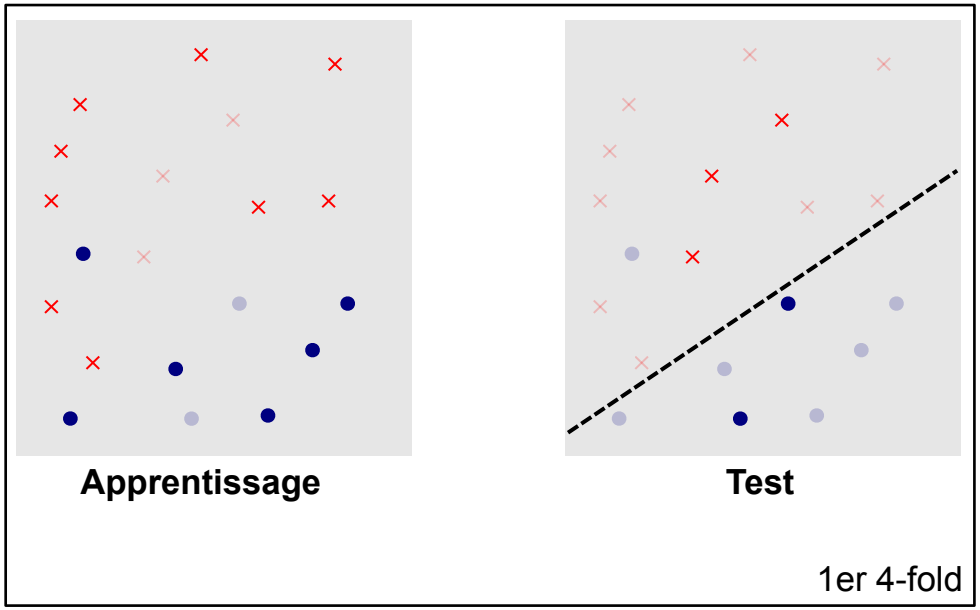


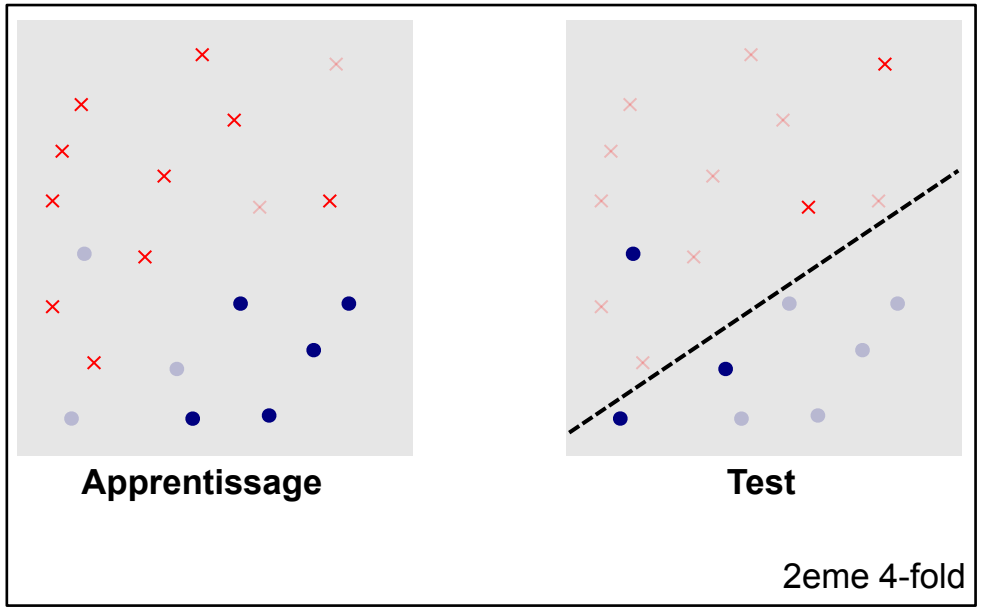
Apprentissage sur toutes les observations sauf 4

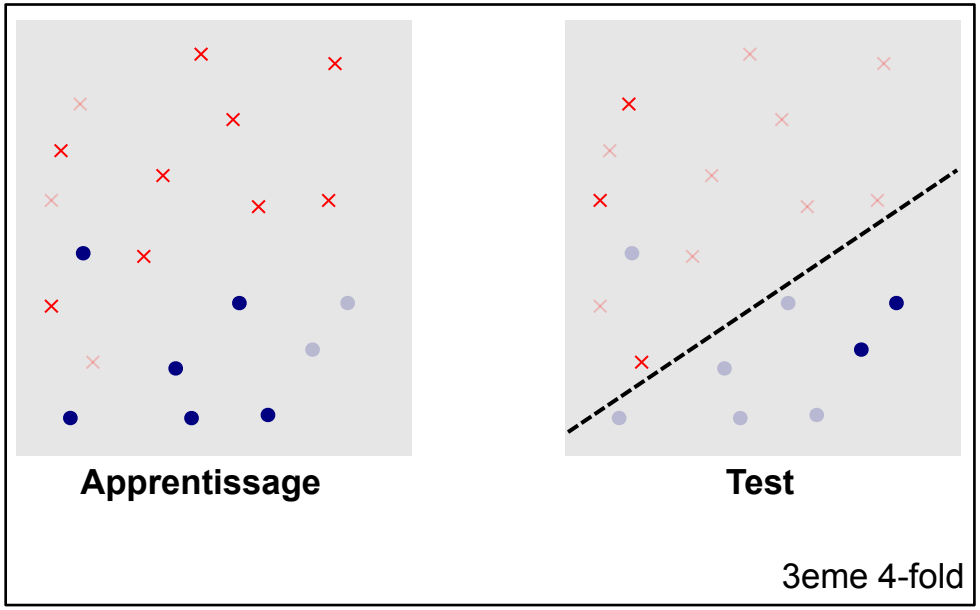


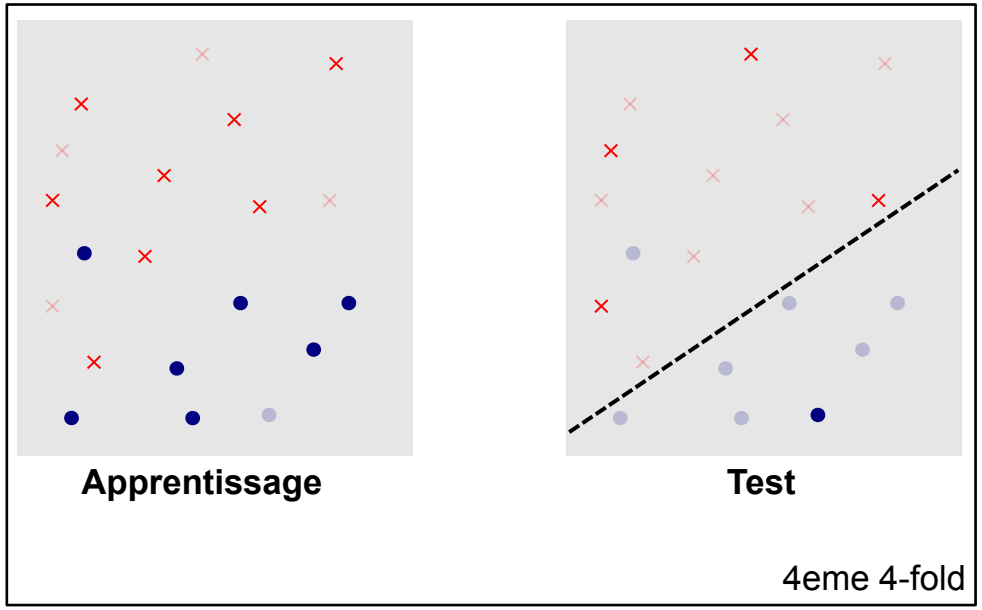
Test sur les 4 données enlevées

K-folds

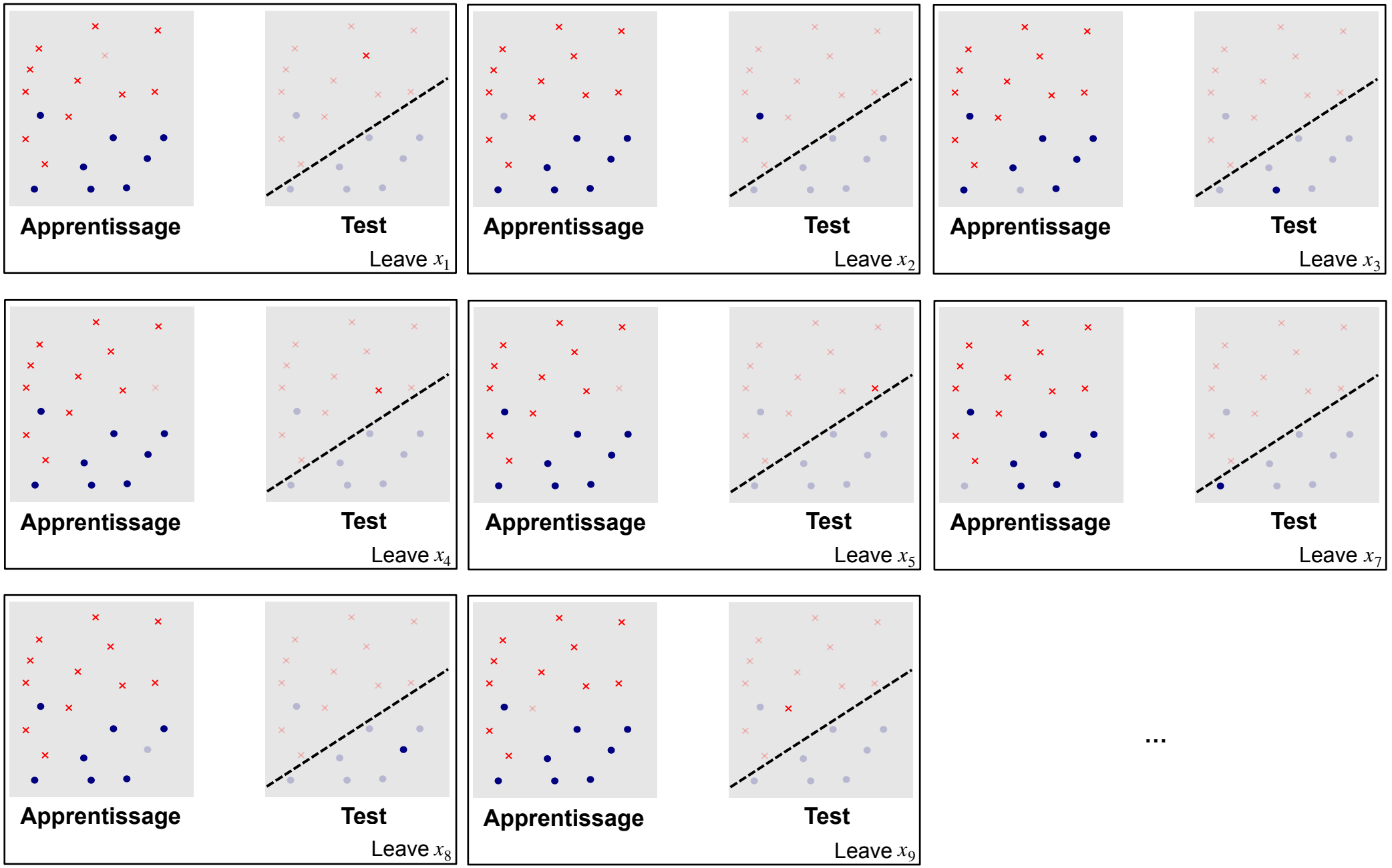

Apprentissage
Test
1er 4-fold


Apprentissage
Test
2eme 4-fold


Apprentissage
Test
3eme 4-fold


Apprentissage
Test
4eme 4-fold

Leave-1-out

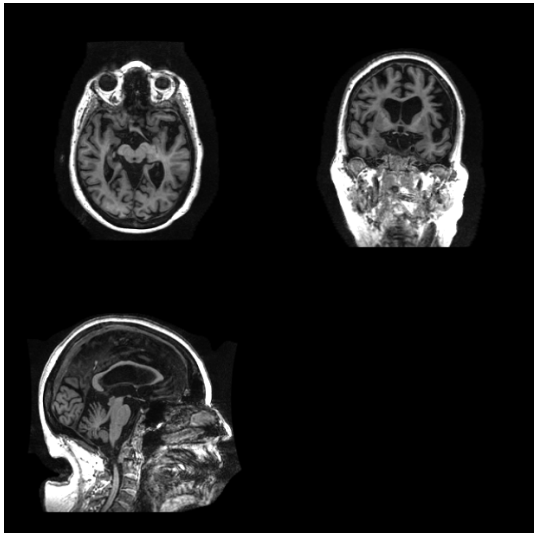

Apprentissage
Test
Leave x_1
Apprentissage
Test
Leave x_2
Apprentissage
Test
Leave x_3
Apprentissage
Test
Leave x_4
Apprentissage
Test
Leave x_5
Apprentissage
Test
Leave x_7
Apprentissage
Test
Leave x_8
Apprentissage
Test
Leave x_9
...

Grande dimension… régularisation et sélection de modèle

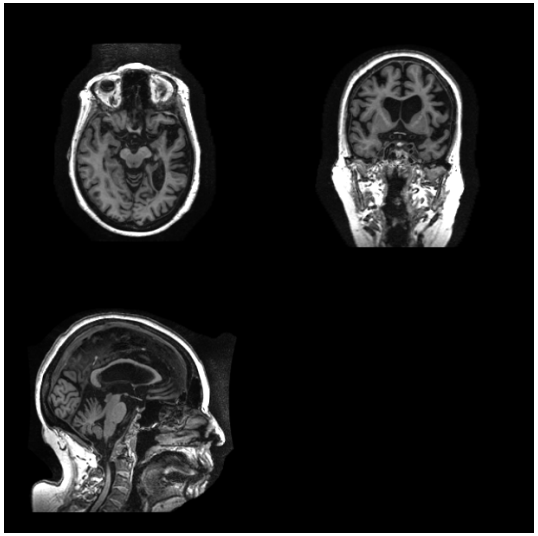
4.a) Grande dimension — exemple de données

Context du projet :

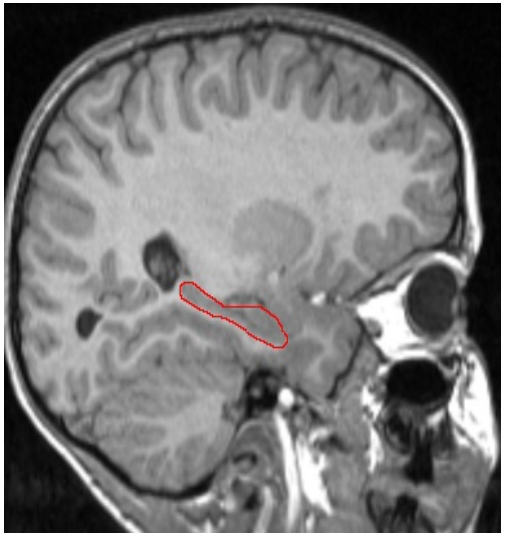
- Observations = IRM du cerveau à différent temps d'acquisition (ADNI*)
- Labels = Etat du patient (MCI/AD)
- Prédiction maladie d'Alheimer en fonction de l'évolution morphométrique de l'hippocampe ?



[Baseline]



[Baseline + 12 mois]



Hippocampe

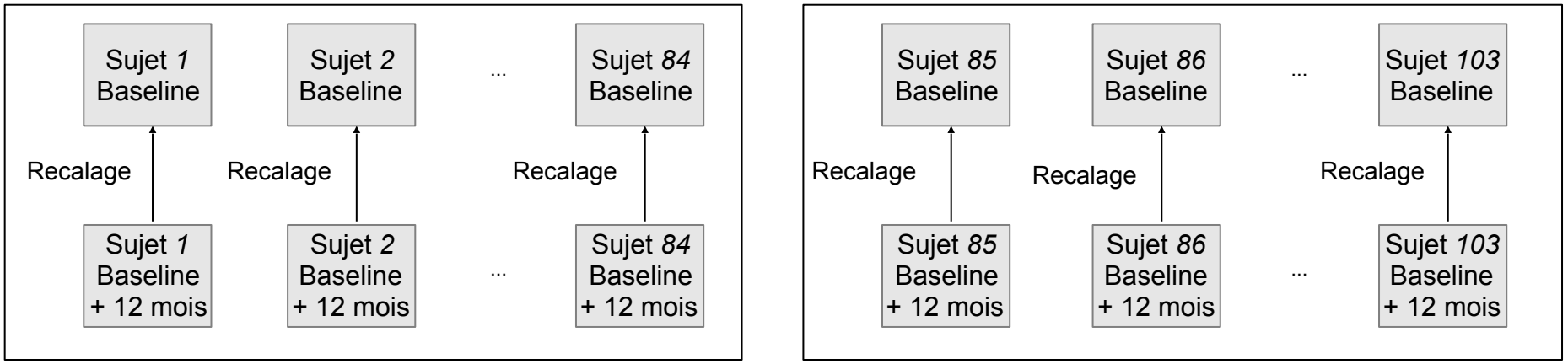
Echantillon d'apprentissage :

- *[Baseline]* : 103 patients sont MCI
- *[Baseline + 12 mois]* : 84 patients sont MCI / 19 patients sont AD

* http://adni.bmap.ucla.edu/

4.a) Grande dimension — exemple de données

Pré-traitement des données d'apprentissage :

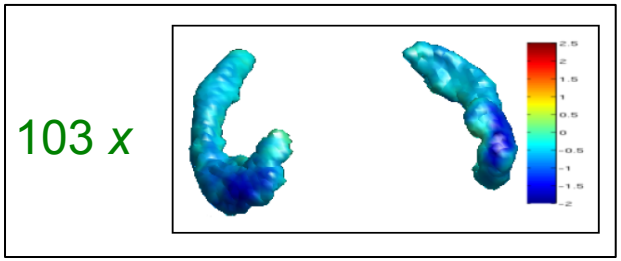


Suivi des déformations et transport des marqueurs d'évolution sur une forme *moyenne* [Vialard et al. IJCV, 2012]

Données d'apprentissage :

Pour chacun des n = 103 sujets :

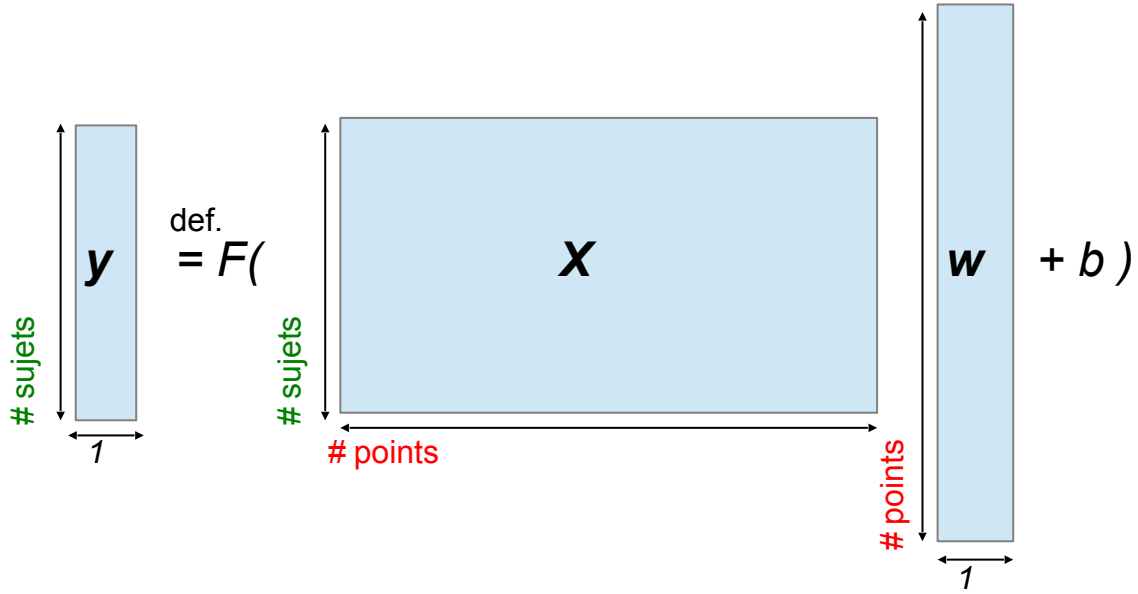
- $\boldsymbol{x_i}$: Observation de l'évolution de la forme sur environ p = 20000 points
- y_i : Etat AD ou MCI



Questions : Discrimination possible ? Quels sont les points les plus discriminants ???

Modèle prédictif de régression logistique qui défini la probabilité des y_i en fonctions des $\boldsymbol{x_i}$:

$$p\left(y_i \mid \mathbf{x}_i, \mathbf{w}, b\right) \overset{\text{def.}}{=} \frac{1}{1+\exp\left(-y_i \ \left(\mathbf{x}_i^T \mathbf{w}+b\right)\right)}$$



Où :

$\boldsymbol{X} \in \boldsymbol{R^{n*p}}$: matrice des $n = 103$ observations de dimension $p = 20000$

$\boldsymbol{y} \in \boldsymbol{\{\mp 1\}^n}$: Etat (*AD = -1 / MCI = 1*)

$\boldsymbol{(w\ ,\ b)} \in \boldsymbol{R^p * R}$: parametres à estimer

Optimisation de la log-vraisemblance :

Paramètre de régularisation obligatoire car p>n

$$\text{Find } (\hat{\mathbf{w}}, \hat{b}) \text{ in } \underset{\mathbf{w},b}{\operatorname{argmin}} \ \mathcal{L}(\mathbf{w}, b) + \lambda J(\mathbf{w})$$

où :

$$\mathcal{L}(\mathbf{w}, b) \overset{\text{def.}}{=} \frac{1}{n} \sum_{i=1}^{n} \log\left(1+\exp\left(-y_i(\mathbf{x}_i^T \mathbf{w}+b)\right)\right)$$

Modèle prédictif de régression logistique qui défini la probabilité des y_i en fonctions des $\mathbf{x}_i$:

$$p\left(y_i \mid \mathbf{x}_i, \mathbf{w}, b\right) \stackrel{\text{def.}}{=} \frac{1}{1+\exp\left(-y_i \ \left(\mathbf{x}_i^T \mathbf{w}+b\right)\right)}$$

$2x = 3$	*n = 1* et *p = 1* — *OK*
$2x_1 + 3x_2 = 3$	*n = 1* et *p = 2* — *KO*
$2x_1 + 3x_2 = 3$ $3x_1 + 1x_2 = 1$	*n = 2* et *p = 2* — *OK*
$2x_1 + 3x_2 + 1x_3 - x_4 = 1$ $5x_1 - x_2 + 2x_3 + x_4 = 1$	*n = 2* et *p = 4* — *KO*

Où :

- $\mathbf{X} \in \mathbb{R}^{n*p}$: matri
- $\mathbf{y} \in \{\mp 1\}^n$: Etat
- $(\mathbf{w}, b) \in \mathbb{R}^p * \mathbb{R}$: parametres à estimer

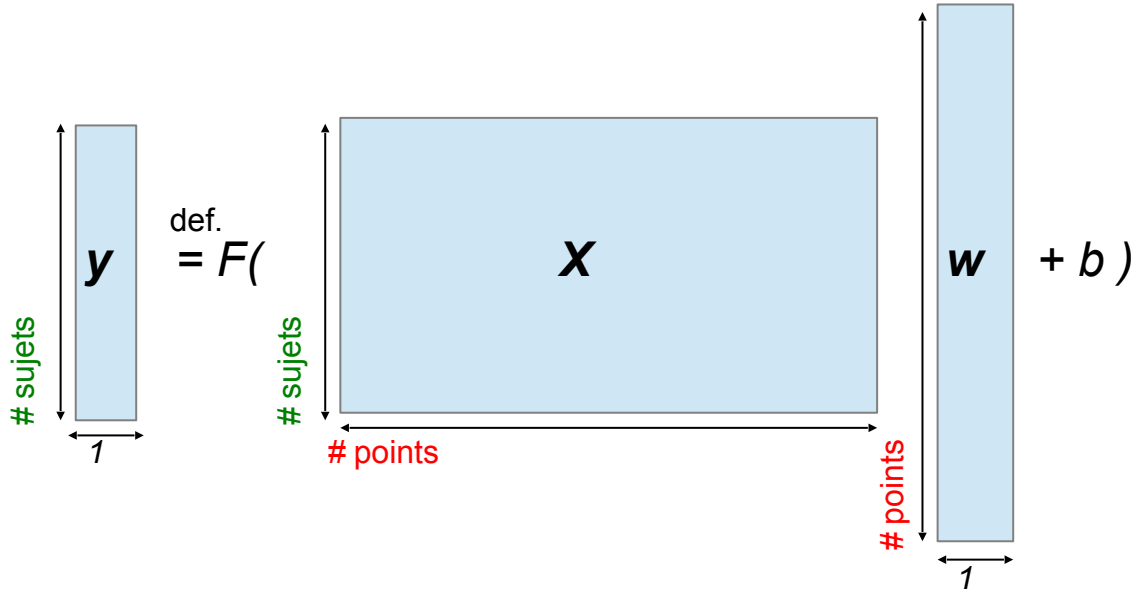
Optimisation de la log-vraisemblance :

Paramètre de régularisation obligatoire car p>n

$$\text{Find } (\hat{\mathbf{w}}, \hat{b}) \text{ in } \underset{\mathbf{w},b}{\operatorname{argmin}} \ \mathcal{L}(\mathbf{w}, b) + \lambda J(\mathbf{w}) \quad \text{où :} \quad \mathcal{L}(\mathbf{w}, b) \stackrel{\text{def.}}{=} \frac{1}{n}\sum_{i=1}^{n} \log\left(1+\exp\left(-y_i(\mathbf{x}_i^T \mathbf{w}+b)\right)\right)$$

Modèle prédictif de régression logistique qui défini la probabilité des y_i en fonctions des $\boldsymbol{x_i}$:

$$p\left(y_i \mid \mathbf{x}_i, \mathbf{w}, b\right) \overset{\text{def.}}{=} \frac{1}{1+\exp\left(-y_i \ \left(\mathbf{x}_i^T \mathbf{w}+b\right)\right)}$$



Où :

$\boldsymbol{X} \in \boldsymbol{R^{n*p}}$: matrice des $n = 103$ observations de dimension $p = 20000$

$\boldsymbol{y} \in \boldsymbol{\{\mp 1\}^n}$: Etat (*AD = -1 / MCI = 1*)

$\boldsymbol{(w\ ,\ b)} \in \boldsymbol{R^p * R}$: parametres à estimer

Optimisation de la log-vraisemblance :

Paramètre de régularisation obligatoire car p>n

$$\text{Find } (\hat{\mathbf{w}}, \hat{b}) \text{ in } \underset{\mathbf{w},b}{\operatorname{argmin}} \ \mathcal{L}(\mathbf{w}, b) + \lambda J(\mathbf{w})$$

où :

$$\mathcal{L}(\mathbf{w}, b) \overset{\text{def.}}{=} \frac{1}{n} \sum_{i=1}^{n} \log\left(1+\exp\left(-y_i(\mathbf{x}_i^T \mathbf{w}+b)\right)\right)$$

$$\text{Find } (\hat{\mathbf{w}}, \hat{b}) \text{ in } \underset{\mathbf{w},b}{\operatorname{argmin}} \ \mathcal{L}(\mathbf{w}, b) + \lambda J(\mathbf{w}) \qquad \text{où :} \qquad \mathcal{L}(\mathbf{w}, b) \overset{\text{def.}}{=} \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + \exp\left(-y_i(\mathbf{x}_i^T \mathbf{w} + b)\right)\right)$$

Exploration de différents modèles de régularisation :

(1) Ridge : $J(\mathbf{w}) = ||\mathbf{w}||_2$

(2) LASSO : $J(\mathbf{w}) = ||\mathbf{w}||_1$

(3) Elastic net : $J(\mathbf{w}) = \lambda_1 ||\mathbf{w}||_1 + \lambda_2 ||\mathbf{w}||_2$

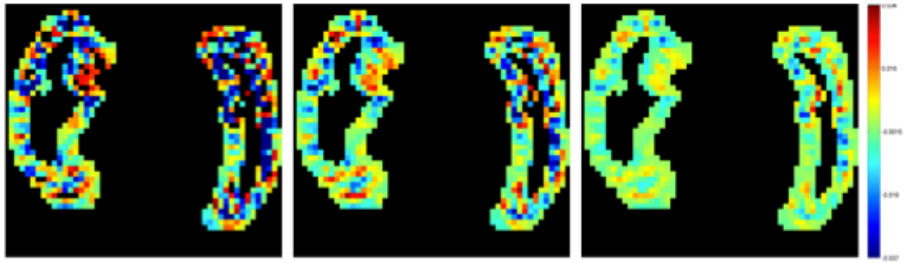
(4) Sobolev semi-norm: $J(\mathbf{w}) = \sum_{\omega \in \mathbf{\Omega}} |\nabla_{\mathbf{\Omega}} \mathbf{w}(\omega)|$

(5) Total Variation : $J(\mathbf{w}) = \left(\sum_{\omega \in \mathbf{\Omega}} |\nabla_{\mathbf{\Omega}} \mathbf{w}(\omega)|^2\right)^{1/2}$

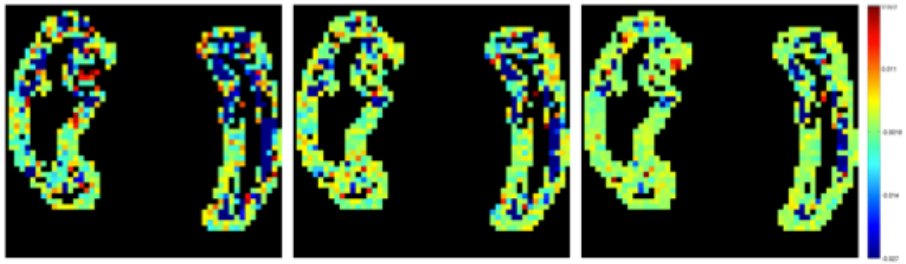
(6) Fused LASSO : $J(\mathbf{w}) = \lambda_1 ||\mathbf{w}||_1 + \lambda_2 \sum_{\omega \in \mathbf{\Omega}} |\nabla_{\mathbf{\Omega}} \mathbf{w}(\omega)|$

Minimisation de la log-vraisemblance en fonction de ***w*** [Lewis & Overton. Math. Programming 2012]

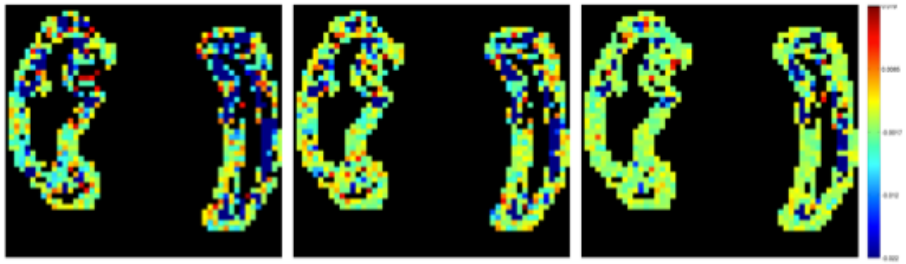
4.c) Grande dimension — Effet de la régularisation



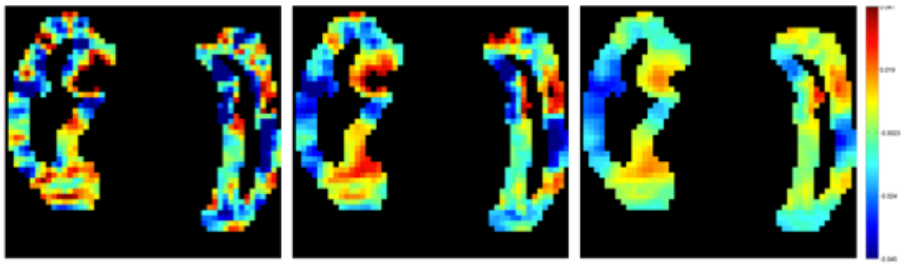
(1) Ridge



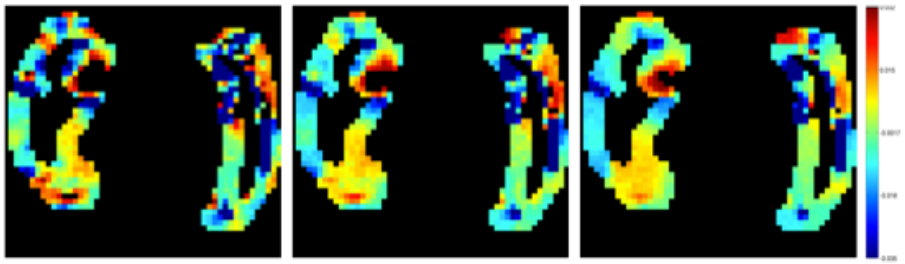
(2) LASSO



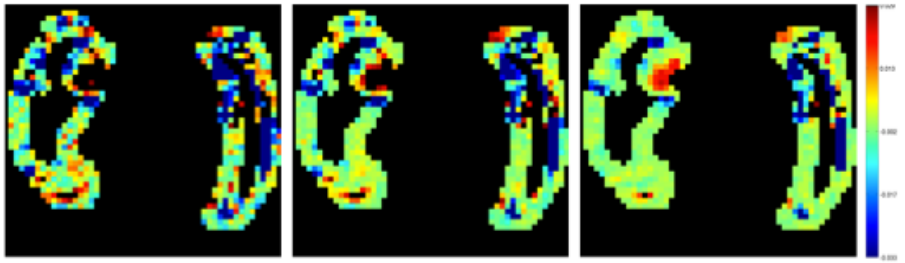
(3) Elastic net



(4) Sobolev semi-norm



(5) Total Variation



(6) Fused LASSO

Représentation de $\boldsymbol{w}$ pour trois λ sur un plan de l'hippocampe :

- Bleu et rouge : forte influence locale
- Vert : peu ou pas d'influence locale

4.c) Grande dimension — Effet de la régularisation

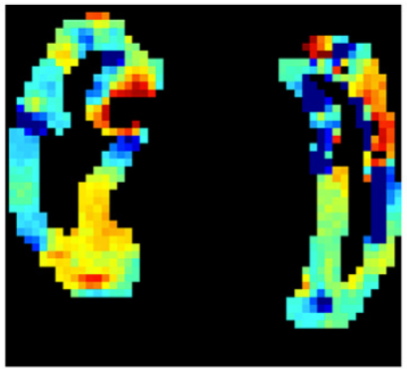
Résultats obtenus avec une méthode de cross validation (ici leave-10%-out) :

- Spec+Sens = 2 ⟶ bonne prédiction dans 100% des cas
- Spec+Sens = 1 ⟶ pile ou face aurait le même pouvoir prédictif

Regularization		**λ range**	**$\hat{\lambda}$ (optimal λ)**	**Spec+ Sens**
None		0	0	1.00
Standard	LASSO	$[10^{-9},10^{0}]$	0.01	1.04
	Ridge	$[10^{-9},10^{0}]$	0.001	1.06
	Elastic Net	$[10^{-9},10^{0}]^2$	$\begin{cases}\hat{\lambda}_1 = 0.01\\ \hat{\lambda}_2 = 1\end{cases}$	1.13
Spatial	Sobolev	$[10^{-9},10^{7}]$	10^{4}	1.17
	Total Variation	$[10^{-9},10^{0}]$	0.01	**1.31**
	Fused LASSO	$[10^{-9},10^{0}]^2$	$\begin{cases}\hat{\lambda}_1 = 0.01\\ \hat{\lambda}_2 = 10^{-4}\end{cases}$	**1.32**

Meilleurs résultats avec une régularisation en pertinente avec les données :

- Tient compte de la distribution spatial
- Permet quelques transitions franches



[Fiot J.B. et al., NeuroImage: Clinical, 2012]

Réduction de dimension par Analyse en Composantes Principales (ACP)

Records nationaux (en secondes) de p = 9 épreuves d'athlétisme pour n = 26 pays

	100m	200m	400m	800m	1500m	5000m	10000m	SemiMarathon	Marathon
Australie	9.93	20.06	44.38	104.40	211.96	775.76	1649.73	3602	7671
Belgique	10.02	20.19	44.78	103.86	214.13	769.71	1612.30	3605	7640
Brésil	10.00	19.89	44.29	101.77	213.25	799.43	1648.12	3573	7565
RoyaumeUni	9.87	19.87	44.36	101.73	209.67	780.41	1638.14	3609	7633
Canada	9.84	20.17	44.44	103.68	211.71	793.96	1656.01	3650	7809
Chine	10.17	20.54	45.25	106.44	216.49	805.14	1670.00	3635	7695
Croatie	10.25	20.76	45.64	104.07	213.30	817.76	1704.32	3827	8225
Ethiopie	10.50	21.08	45.89	106.08	211.13	757.35	1577.53	3535	7439
France	9.99	20.16	44.46	103.15	208.98	778.83	1642.78	3658	7596
Allemagne	10.06	20.20	44.33	103.65	211.58	774.70	1641.53	3634	7727
Inde	10.30	20.73	45.48	105.77	218.00	809.70	1682.89	3672	7920
Iran	10.29	21.11	46.37	104.74	218.80	833.40	1762.65	4103	8903
Italie	10.01	19.72	45.19	103.17	212.78	785.59	1636.50	3620	7642
Jamaïque	9.58	19.19	44.49	105.21	219.19	813.10	1712.44	3816	8199
Japon	10.00	20.03	44.78	106.18	217.42	793.20	1655.09	3625	7576
Kenya	10.26	20.43	44.18	102.01	206.34	759.74	1587.85	3513	7467
Lituanie	10.33	20.88	45.73	106.64	220.90	797.90	1651.50	3851	7955
NouvelleZélande	10.11	20.42	46.09	104.30	212.17	790.19	1661.95	3732	7815
Portugal	9.86	20.01	46.11	104.91	210.07	782.86	1632.47	3665	7596
Russie	10.10	20.23	44.60	102.47	212.28	791.99	1673.12	3675	7747
AfriqueduSud	10.06	20.11	44.59	102.69	213.56	794.16	1649.94	3678	7593
Espagne	10.14	20.59	44.96	103.83	208.95	782.54	1634.44	3592	7562
Suède	10.18	20.30	44.56	105.54	216.49	797.59	1675.74	3655	7838
Suisse	10.16	20.41	44.99	102.55	211.75	787.54	1673.16	3686	7643
Ukraine	10.07	20.00	45.11	105.08	210.33	790.78	1679.80	3711	7635
USA	9.69	19.32	43.18	102.60	209.30	776.27	1633.98	3583	7538

Comment faire un classement général entre les pays ???

⟶ Somme pondérée des scores, puis classement en fonction du rang.

5) Réduction de dimension par ACP

Records nationaux (en secondes) de p = 9 épreuves d'athlétisme pour n = 26 pays

	100m	200m	400m	800m	1500m	5000m	10000m	SemiMarathon	Marathon
Australie	9.93	20.06	44.38	104.40	211.96	775.76	1649.73	3602	7671
Belgique	10.02	20.19	44.78	103.86	214.13	769.71	1612.30	3605	7640
Brésil	10.00	19.89	44.29	101.77	213.25	799.43	1648.12	3573	7565
RoyaumeUni	9.87	19.87	44.36	101.73	209.67	780.41	1638.14	3609	7633
Canada	9.84	20.17	44.44	103.68	211.71	793.96	1656.01	3650	7809
Chine	10.17	20.54	45.25	106.44	216.49	805.14	1670.00	3635	7695
Croatie	10.25	20.76	45.64	104.07	213.30	817.76	1704.32	3827	8225
Ethiopie	10.50	21.08	45.89	106.08	211.13	757.35	1577.53	3535	7439
France	9.99	20.16	44.46	103.15	208.98	778.83	1642.78	3658	7596
Allemagne	10.06	20.20	44.33	103.65	211.58	774.70	1641.53	3634	7727
Inde	10.30	20.73	45.48	105.77	218.00	809.70	1682.89	3672	7920
Iran	10.29	21.11	46.37	104.74	218.80	833.40	1762.65	4103	8903
Italie	10.01	19.72	45.19	103.17	212.78	785.59	1636.50	3620	7642
Jamaïque	9.58	19.19	44.49	105.21	219.19	813.10	1712.44	3816	8199
Japon	10.00	20.03	44.78	106.18	217.42	793.20	1655.09	3625	7576
Kenya	10.26	20.43	44.18	102.01	206.34	759.74	1587.85	3513	7467
Lituanie	10.33	20.88	45.73	106.64	220.90	797.90	1651.50	3851	7955
NouvelleZélande	10.11	20.42	46.09	104.30	212.17	790.19	1661.95	3732	7815
Portugal	9.86	20.01	46.11	104.91	210.07	782.86	1632.47	3665	7596
Russie	10.10	20.23	44.60	102.47	212.28	791.99	1673.12	3675	7747
AfriqueduSud	10.06	20.11	44.59	102.69	213.56	794.16	1649.94	3678	7593
Espagne	10.14	20.59	44.96	103.83	208.95	782.54	1634.44	3592	7562
Suède	10.18	20.30	44.56	105.54	216.49	797.59	1675.74	3655	7838
Suisse	10.16	20.41	44.99	102.55	211.75	787.54	1673.16	3686	7643
Ukraine	10.07	20.00	45.11	105.08	210.33	790.78	1679.80	3711	7635
USA	9.69	19.32	43.18	102.60	209.30	776.27	1633.98	3583	7538

→ Matrice ***M***

Comment faire un classement général entre les pays ???

⟶ Somme pondérée des scores, puis classement en fonction du rang.

Somme pondérée des scores est équivalente à une multiplication matrice *x* vecteur :

⟶ Vecteur contenant les scores = $\boldsymbol{M} \cdot \boldsymbol{w}$

5) Réduction de dimension par ACP

Records nationaux (en secondes) de p = 9 épreuves d'athlétisme pour n = 26 pays

	100m	200m	400m	800m	1500m	5000m	10000m	SemiMarathon	Marathon
Australie	9.93	20.06	44.38	104.40	211.96	775.76	1649.73	3602	7671
Belgique	10.02	20.19	44.78	103.86	214.13	769.71	1612.30	3605	7640
Brésil	10.00	19.89	44.29	101.77	213.25	799.43	1648.12	3573	7565
RoyaumeUni	9.87	19.87	44.36	101.73	209.67	780.41	1638.14	3609	7633
Canada	9.84	20.17	44.44	103.68	211.71	793.96	1656.01	3650	7809
Chine	10.17	20.54	45.25	106.44	216.49	805.14	1670.00	3635	7695
Croatie	10.25	20.76	45.64	104.07	213.30	817.76	1704.32	3827	8225
Ethiopie	10.50	21.08	45.89	106.08	211.13	757.35	1577.53	3535	7439
France	9.99	20.16	44.46	103.15	208.98	778.83	1642.78	3658	7596
Allemagne	10.06	20.20	44.33	103.65	211.58	774.70	1641.53	3634	7727
Inde	10.30	20.73	45.48	105.77	218.00	809.70	1682.89	3672	7920
Iran	10.29	21.11	46.37	104.74	218.80	833.40	1762.65	4103	8903
Italie	10.01	19.72	45.19	103.17	212.78	785.59	1636.50	3620	7642
Jamaïque	9.58	19.19	44.49	105.21	219.19	813.10	1712.44	3816	8199
Japon	10.00	20.03	44.78	106.18	217.42	793.20	1655.09	3625	7576
Kenya	10.26	20.43	44.18	102.01	206.34	759.74	1587.85	3513	7467
Lituanie	10.33	20.88	45.73	106.64	220.90	797.90	1651.50	3851	7955
NouvelleZélande	10.11	20.42	46.09	104.30	212.17	790.19	1661.95	3732	7815
Portugal	9.86	20.01	46.11	104.91	210.07	782.86	1632.47	3665	7596
Russie	10.10	20.23	44.60	102.47	212.28	791.99	1673.12	3675	7747
AfriqueduSud	10.06	20.11	44.59	102.69	213.56	794.16	1649.94	3678	7593
Espagne	10.14	20.59	44.96	103.83	208.95	782.54	1634.44	3592	7562
Suède	10.18	20.30	44.56	105.54	216.49	797.59	1675.74	3655	7838
Suisse	10.16	20.41	44.99	102.55	211.75	787.54	1673.16	3686	7643
Ukraine	10.07	20.00	45.11	105.08	210.33	790.78	1679.80	3711	7635
USA	9.69	19.32	43.18	102.60	209.30	776.27	1633.98	3583	7538

→ Matrice $\boldsymbol{M}$

On peut aussi chercher le vecteur (de norme 1) qui maximise la variabilité entre les scores

⟶ Vecteur optimal = 1er vecteur propre ($\boldsymbol{v_1}$) de l'ACP

⟶ Niveau de variabilité = 1ere valeur propre (λ_1) de l'ACP

⟶ Vecteur de scores avec la plus grande variabilité possible = $\boldsymbol{M} \cdot \boldsymbol{v_1}$

5) Réduction de dimension par ACP

Records nationaux (en secondes) de p = 9 épreuves d'athlétisme pour n = 26 pays

	100m	200m	400m	800m	1500m	5000m	10000m	SemiMarathon	Marathon
Australie	9.93	20.06	44.38	104.40	211.96	775.76	1649.73	3602	7671
Belgique	10.02	20.19	44.78	103.86	214.13	769.71	1612.30	3605	7640
Brésil	10.00	19.89	44.29	101.77	213.25	799.43	1648.12	3573	7565
RoyaumeUni	9.87	19.87	44.36	101.73	209.67	780.41	1638.14	3609	7633
Canada	9.84	20.17	44.44	103.68	211.71	793.96	1656.01	3650	7809
Chine	10.17	20.54	45.25	106.44	216.49	805.14	1670.00	3635	7695
Croatie	10.25	20.76	45.64	104.07	213.30	817.76	1704.32	3827	8225
Ethiopie	10.50	21.08	45.89	106.08	211.13	757.35	1577.53	3535	7439
France	9.99	20.16	44.46	103.15	208.98	778.83	1642.78	3658	7596
Allemagne	10.06	20.20	44.33	103.65	211.58	774.70	1641.53	3634	7727
Inde	10.30	20.73	45.48	105.77	218.00	809.70	1682.89	3672	7920
Iran	10.29	21.11	46.37	104.74	218.80	833.40	1762.65	4103	8903
Italie	10.01	19.72	45.19	103.17	212.78	785.59	1636.50	3620	7642
Jamaïque	9.58	19.19	44.49	105.21	219.19	813.10	1712.44	3816	8199
Japon	10.00	20.03	44.78	106.18	217.42	793.20	1655.09	3625	7576
Kenya	10.26	20.43	44.18	102.01	206.34	759.74	1587.85	3513	7467
Lituanie	10.33	20.88	45.73	106.64	220.90	797.90	1651.50	3851	7955
NouvelleZélande	10.11	20.42	46.09	104.30	212.17	790.19	1661.95	3732	7815
Portugal	9.86	20.01	46.11	104.91	210.07	782.86	1632.47	3665	7596
Russie	10.10	20.23	44.60	102.47	212.28	791.99	1673.12	3675	7747
AfriqueduSud	10.06	20.11	44.59	102.69	213.56	794.16	1649.94	3678	7593
Espagne	10.14	20.59	44.96	103.83	208.95	782.54	1634.44	3592	7562
Suède	10.18	20.30	44.56	105.54	216.49	797.59	1675.74	3655	7838
Suisse	10.16	20.41	44.99	102.55	211.75	787.54	1673.16	3686	7643
Ukraine	10.07	20.00	45.11	105.08	210.33	790.78	1679.80	3711	7635
USA	9.69	19.32	43.18	102.60	209.30	776.27	1633.98	3583	7538

→ Matrice ***M***

Une fois enlevée l'influence de v_1, on cherche le vecteur (de norme 1) qui maximise la variabilité

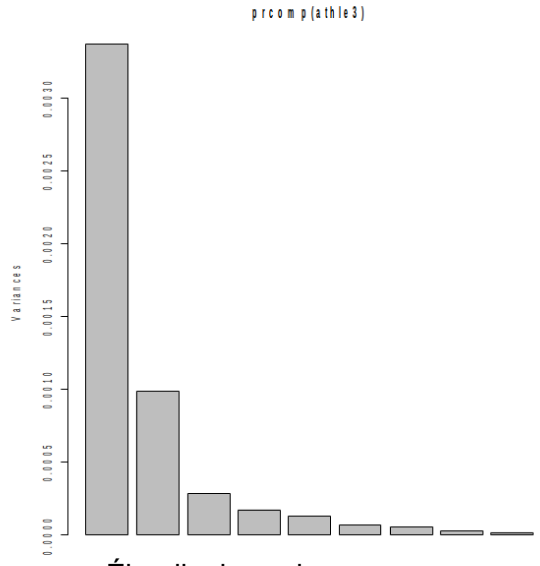
⟶ Vecteur optimal = 2er vecteur propre ($\boldsymbol{v_2}$) de l'ACP

⟶ Niveau de variabilité = 2ere valeur propre (λ_2) de l'ACP

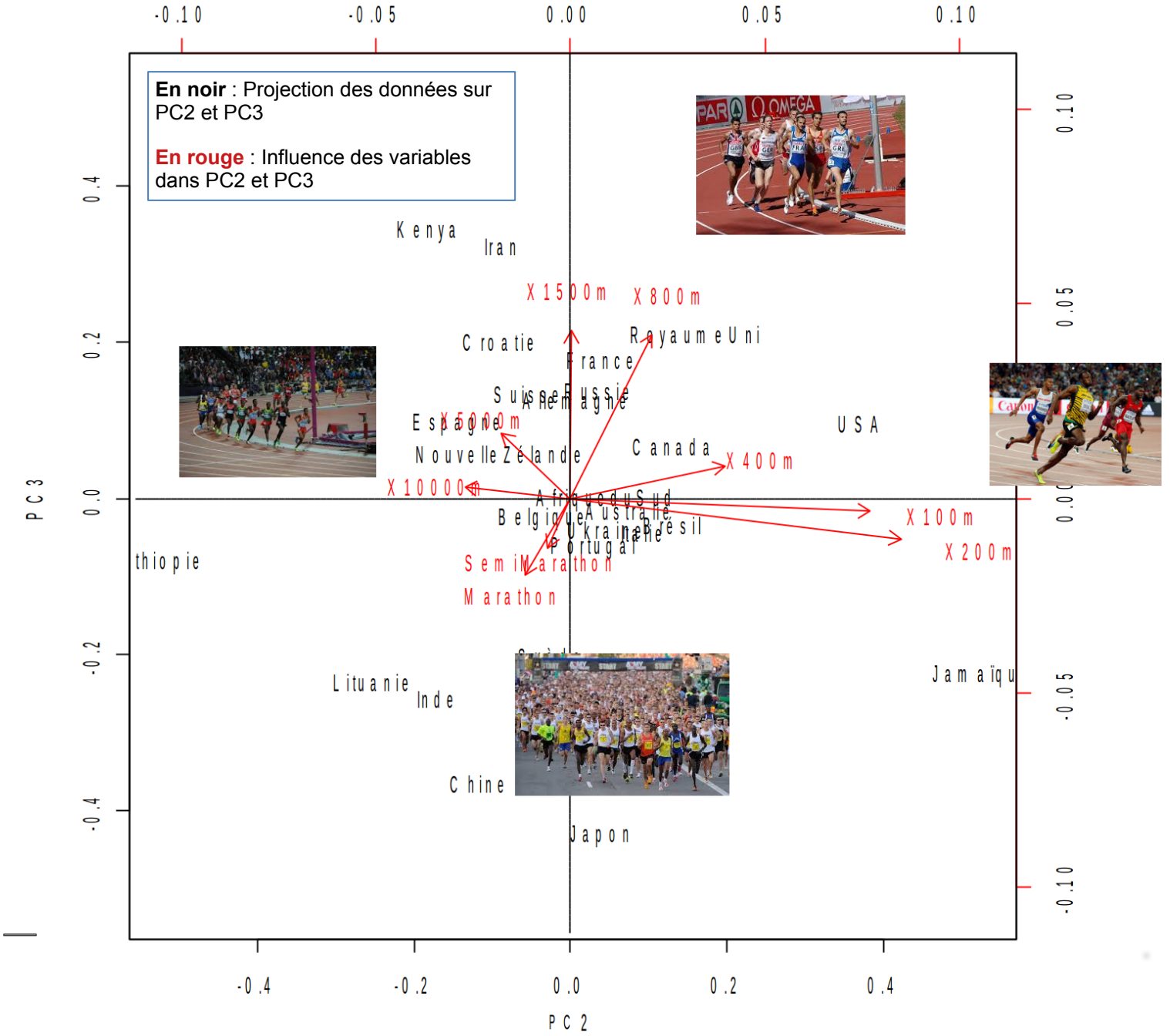
…

Calculable de manière analytique

5) Réduction de dimension par ACP



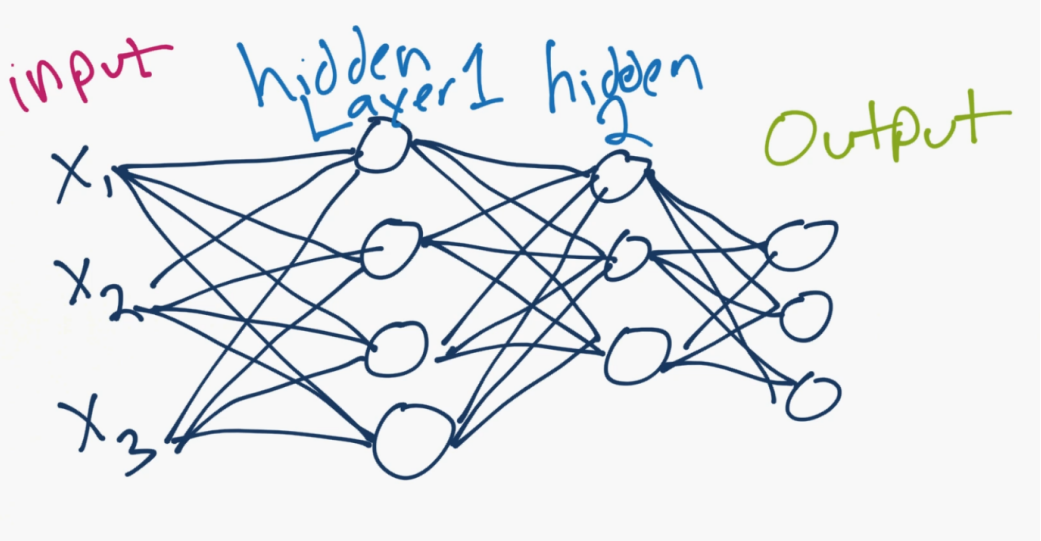
Éboulis des valeurs propres



Apprentissage machine et GPU

Deep learning Caffe | Keras | theano | PYTORCH | TensorFlow | …

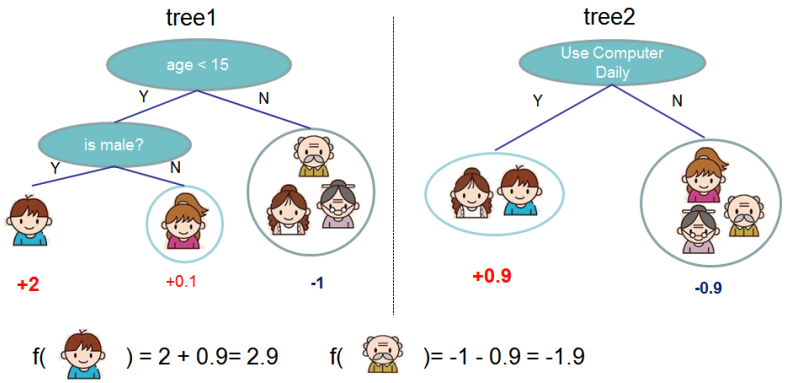
Connu — Appris — Connu



https://pythonprogramming.net/neural-networks-machine-learning-tutorial/

- Très efficace sur certains problèmes (signaux, images).
- Apprentissage nécessitant du calcul intensif mais prédiction rapide.
- Dimension du problème d'apprentissage très élevée.
- Nécessité d'avoir énormément de données annotées ou une structure de graphe adaptée aux données.

XGBoost XGBoost



- Méthode basée sur des arbres de décision.
- Très efficace dans le cas général.
- Paramètre à gérer (contrairement à Random Forest) et gros coût calculatoire.

Prédiction

$\boldsymbol{I}$ =
Image RGB 200*200

Image de chien ou chat

Classifieur (boite noire)

$h^1(\boldsymbol{I})$

$h^2(\boldsymbol{I})$

Prédiction

- Si chien : $h^1(\boldsymbol{I}) == \boldsymbol{0}$
- Si chat : $h^1(\boldsymbol{I}) == \boldsymbol{1}$
- Si gentil : $h^2(\boldsymbol{I}) == \boldsymbol{0}$
- Si agressif : $h^2(\boldsymbol{I}) == \boldsymbol{1}$

Prédiction

Image de chien ou chat

Classifieur (boite noire)

1

0

- Si chien : $h^1(I) == 0$
- Si chat : $h^1(I) == 1$
- Si gentil : $h^2(I) == 0$
- Si agressif : $h^2(I) == 1$

Prédiction

Image de chien ou chat

Classifieur (boite noire)

1

1

- Si chien : $h^1(I) == 0$
- Si chat : $h^1(I) == 1$
- Si gentil : $h^2(I) == 0$
- Si agressif : $h^2(I) == 1$

Prédiction

Image de chien ou chat

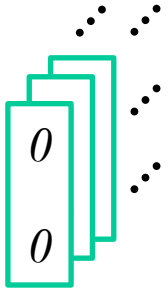
Classifieur (boite noire)

$$\begin{matrix} 0 \\ 0 \end{matrix}$$

- Si chien : $h^1(\boldsymbol{I}) == \boldsymbol{0}$
- Si chat : $h^1(\boldsymbol{I}) == \boldsymbol{1}$
- Si gentil : $h^2(\boldsymbol{I}) == \boldsymbol{0}$
- Si agressif : $h^2(\boldsymbol{I}) == \boldsymbol{1}$

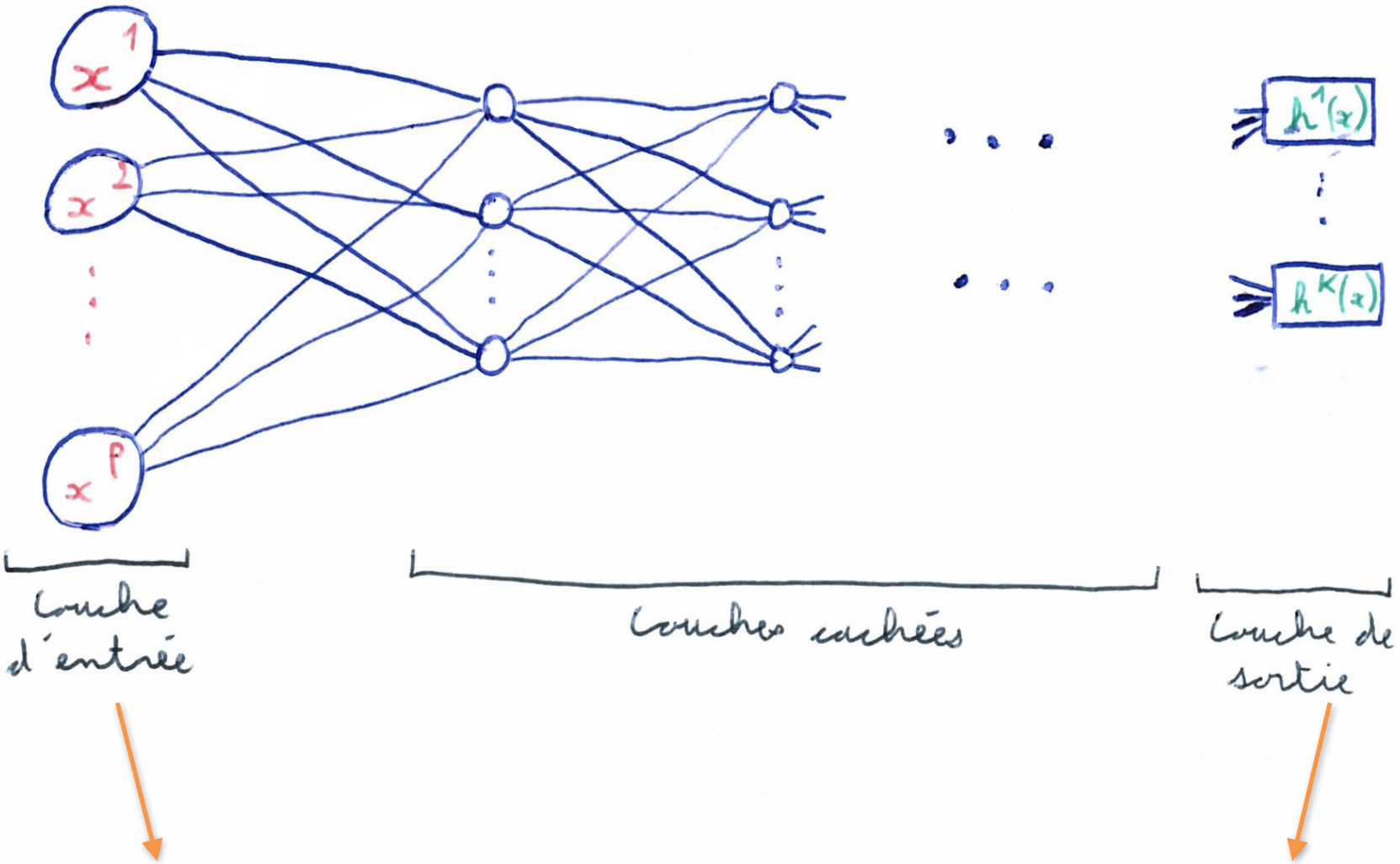
Apprentissage : Optimisation des paramètres pour obtenir les meilleurs prédictions en moyenne

Classifieur (*boite noire*)



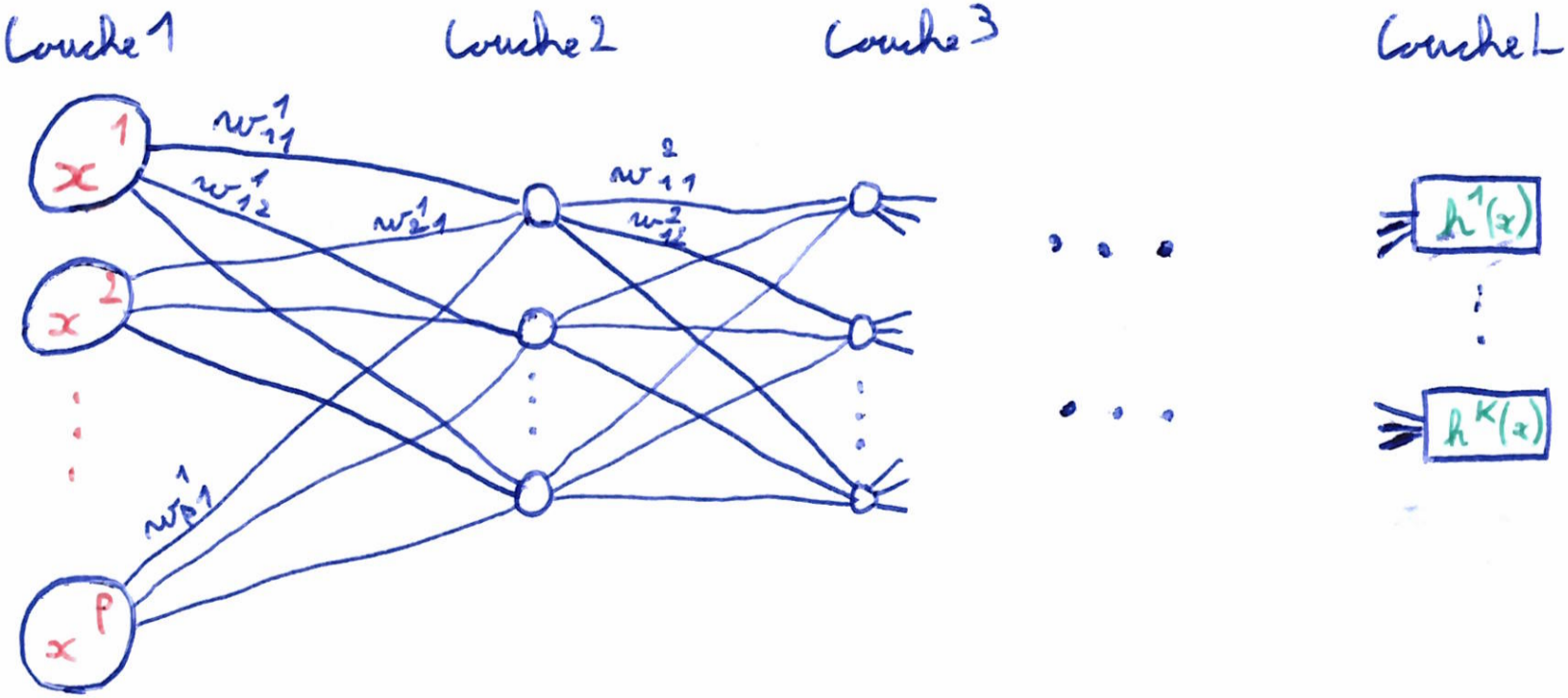
Données d'apprentissage (1/2)
Beaucoup d'images de chien ou chat

Données d'apprentissage (2/2)
Labels de chaque image

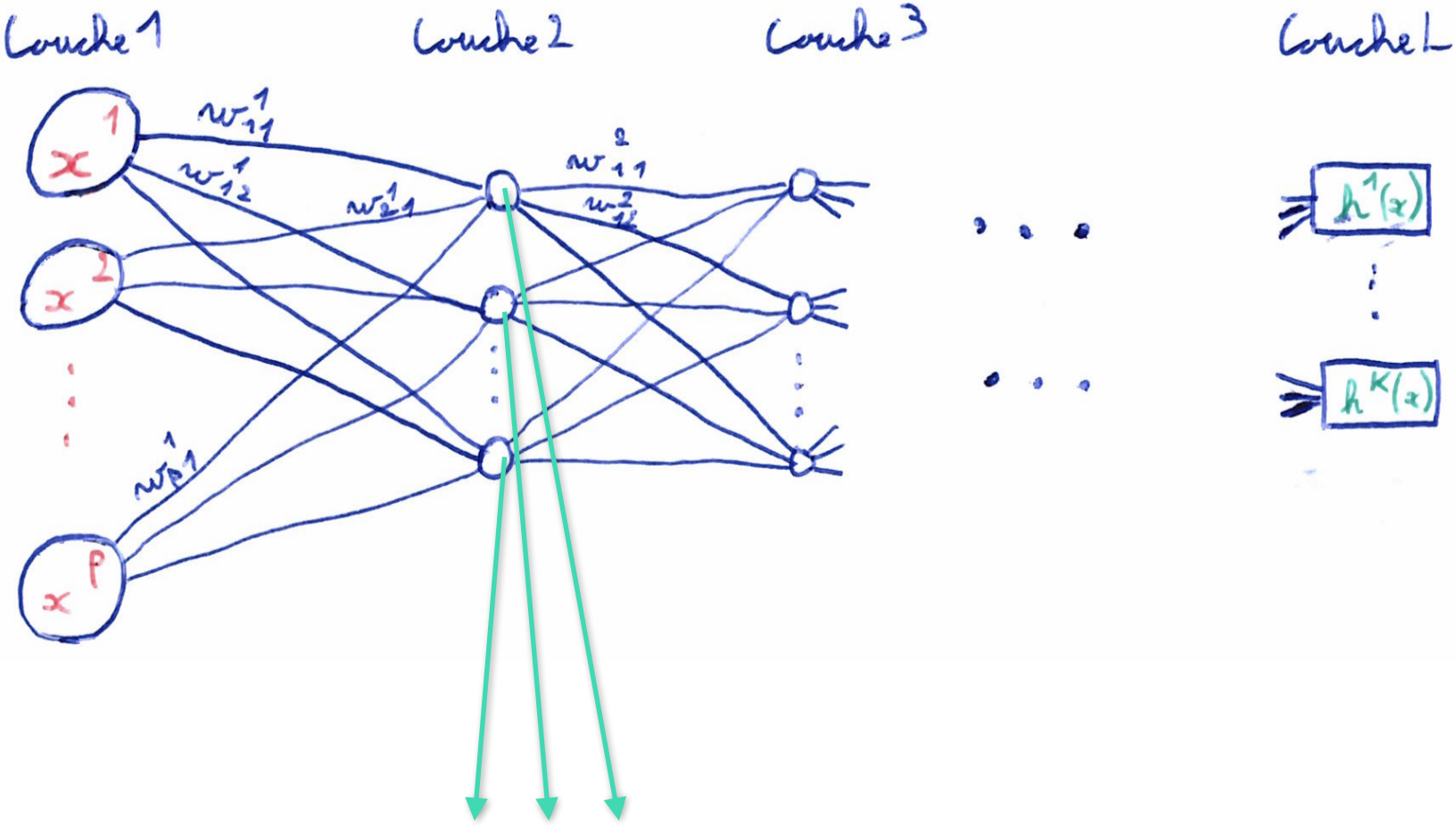


Les x^i peuvent être les intensités de l'image I sur chaque canal RGB

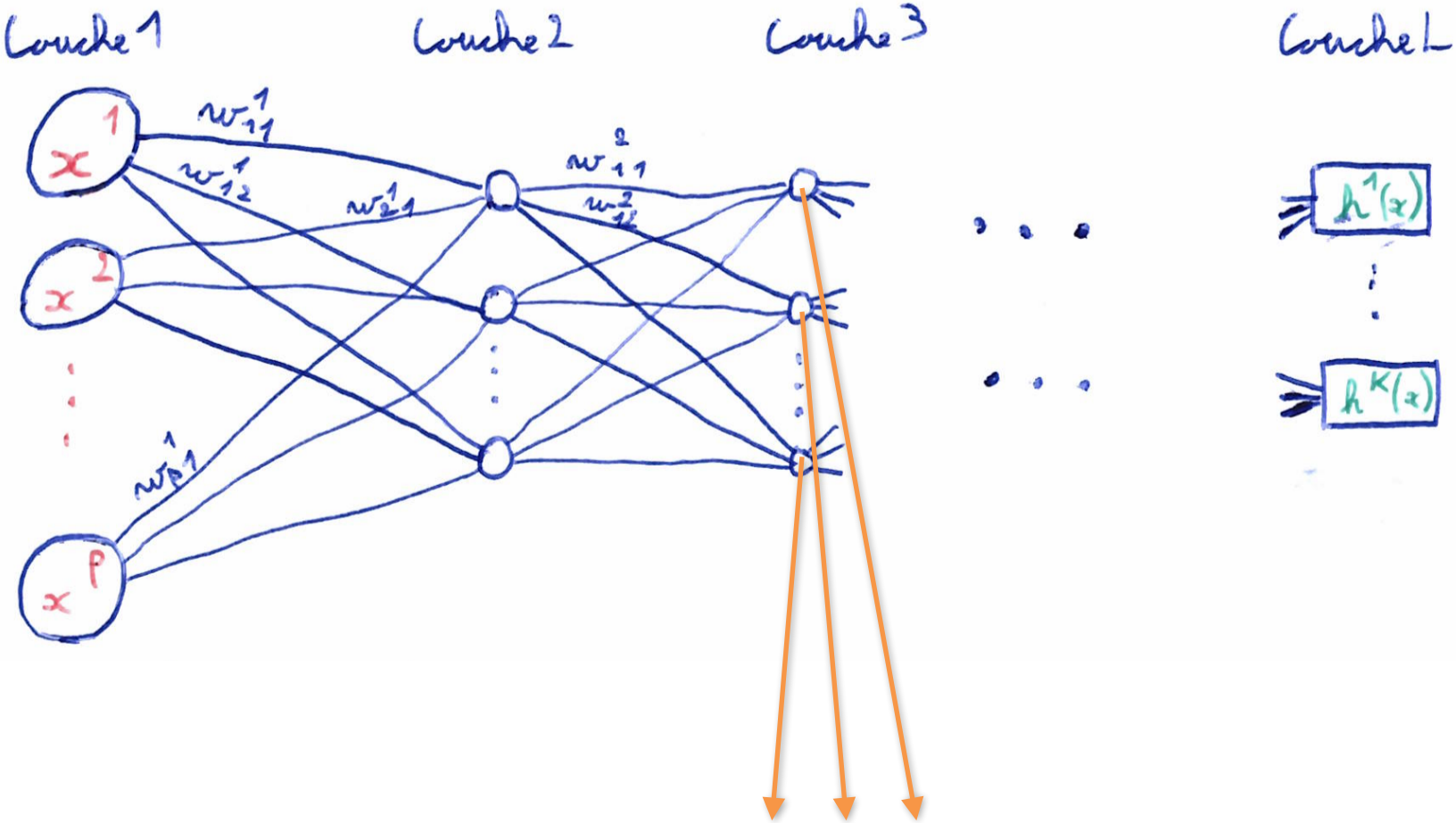
Labels prédits



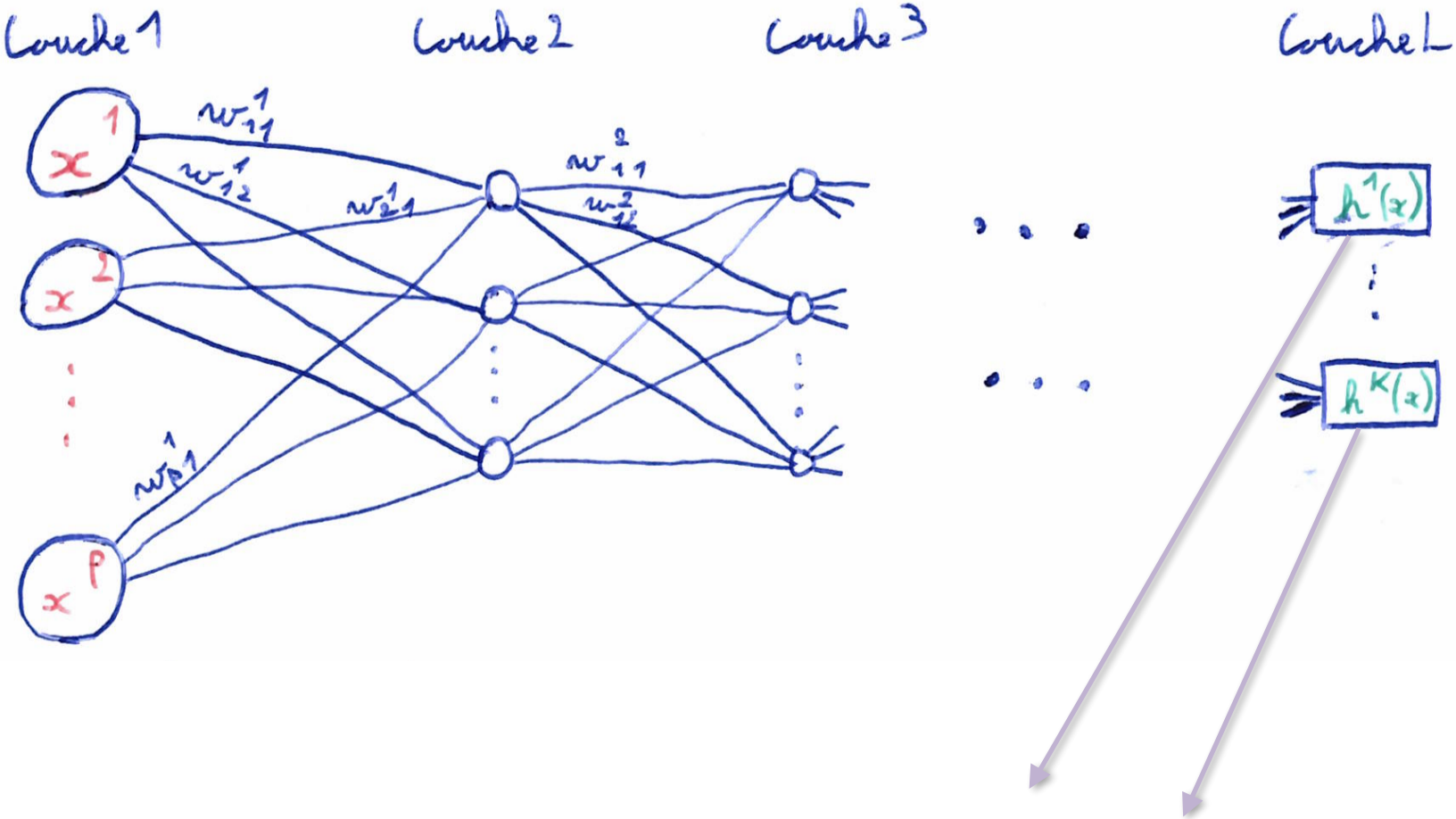
$$\sigma_k^\ell = f(s_k^\ell) = f\left(\sum_{p \in N_{\ell-1}} w_{pk}^{\ell-1} \, \sigma_p^{\ell-1}\right)$$



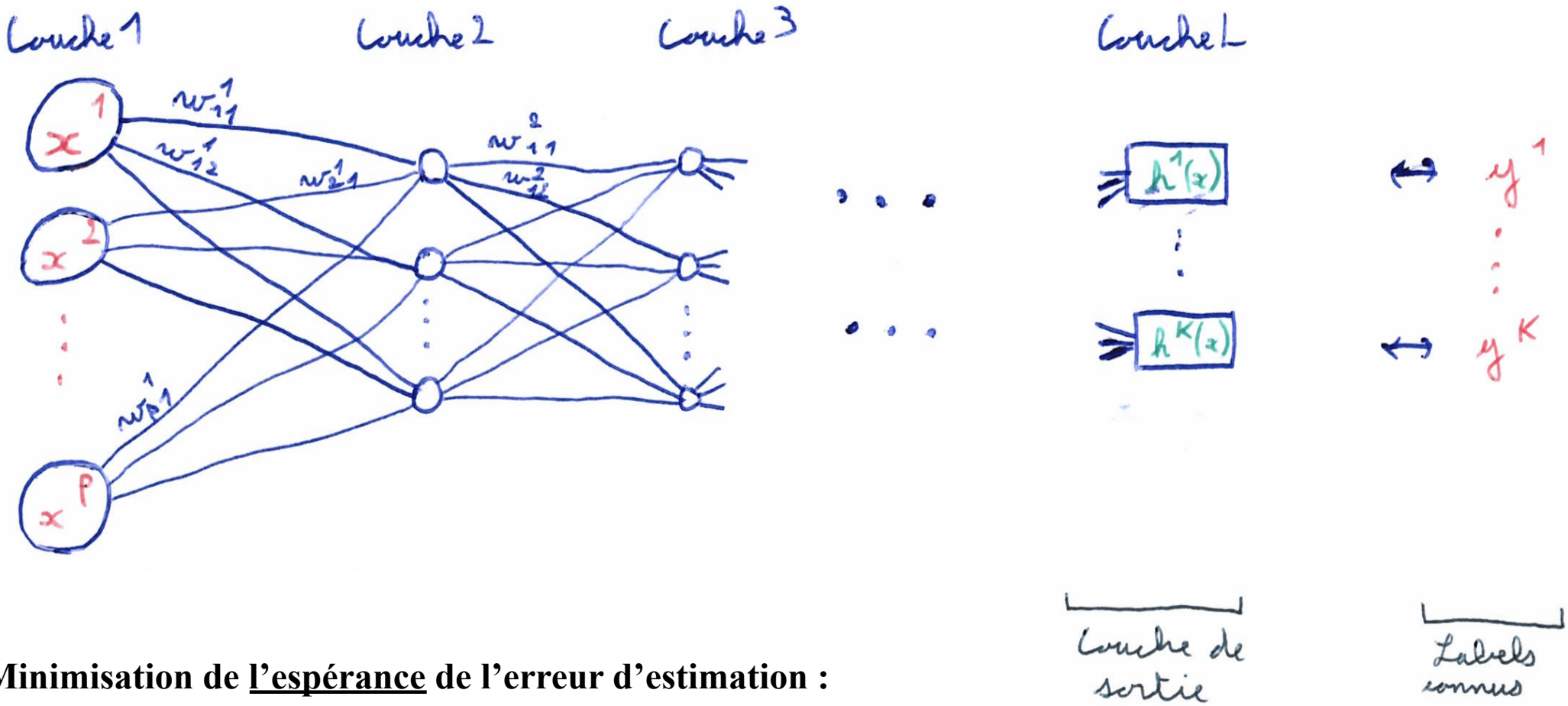
$$\sigma_k^\ell = f(s_k^\ell) = f\left(\sum_{p \in N_{\ell-1}} w_{pk}^{\ell-1} \, \sigma_p^{\ell-1}\right)$$



$$\sigma_k^\ell = f(s_k^\ell) = f\left(\sum_{p \in N_{\ell-1}} w_{pk}^{\ell-1} \, \sigma_p^{\ell-1}\right)$$

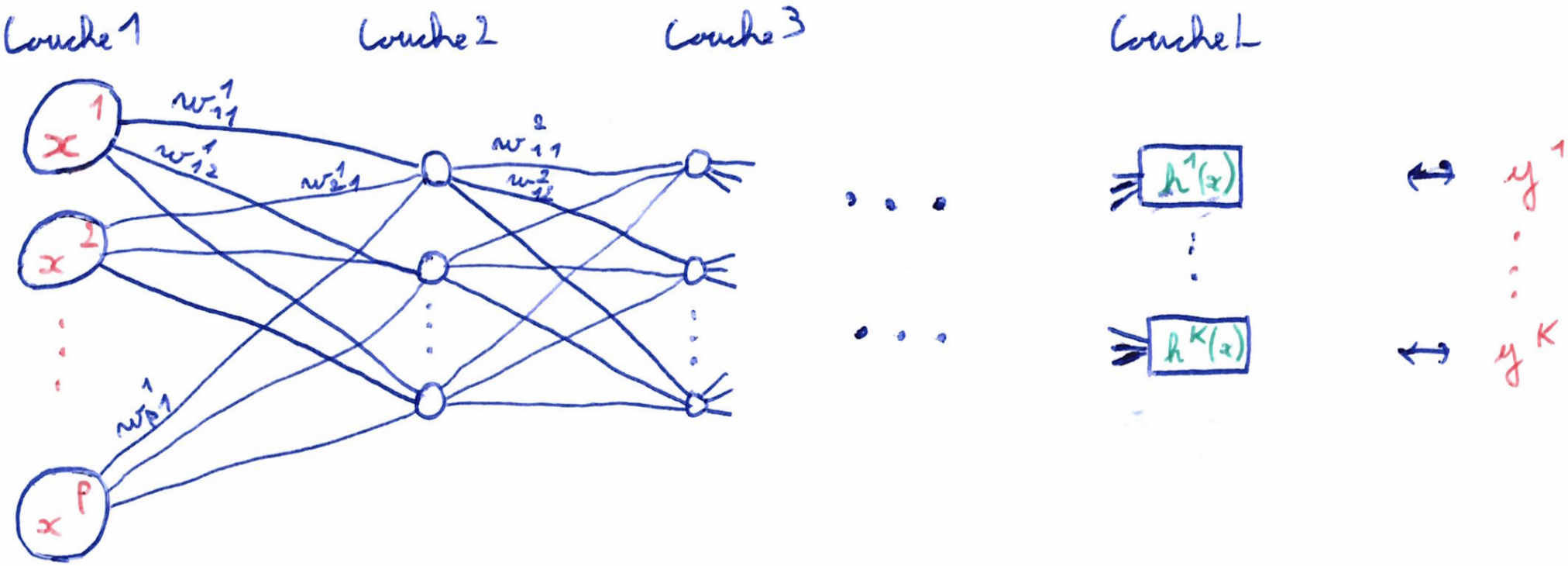


$$\sigma_k^\ell = f(s_k^\ell) = f\left(\sum_{p \in N_{\ell-1}} w_{pk}^{\ell-1} \, \sigma_p^{\ell-1}\right)$$



Minimisation de l'espérance de l'erreur d'estimation :

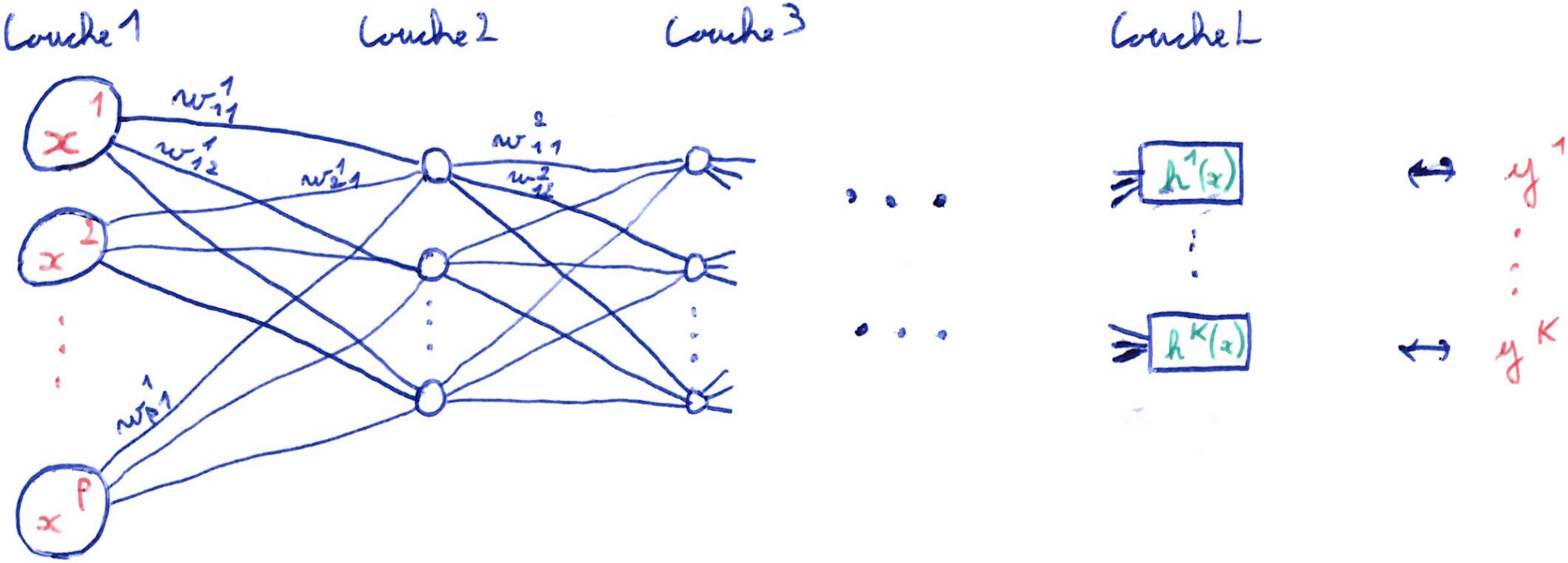
$$E\left(\underbrace{x^1, \dots, x^p}_{x}\right) = \sum_{k=1}^{K} \left(h^k(x) - y^k\right)^2$$



Descente de gradient stochastique :

$$w^l_{ph} \leftarrow w^l_{ph} - \eta \frac{\partial E}{\partial w^l_{ph}} \quad \text{avec} \quad \frac{\partial E}{\partial w^l_{ph}}$$

- Facile à calculer analytiquement pour l=L-1
- Calculable analytiquement pour l<L-1 si connu à l+1
- Stochastique car le gradient de l'espérance est estimé avec un sous-ensemble des observations (mini-batch, online training, …)



Descente de gradient stochastique :

A chaque itération

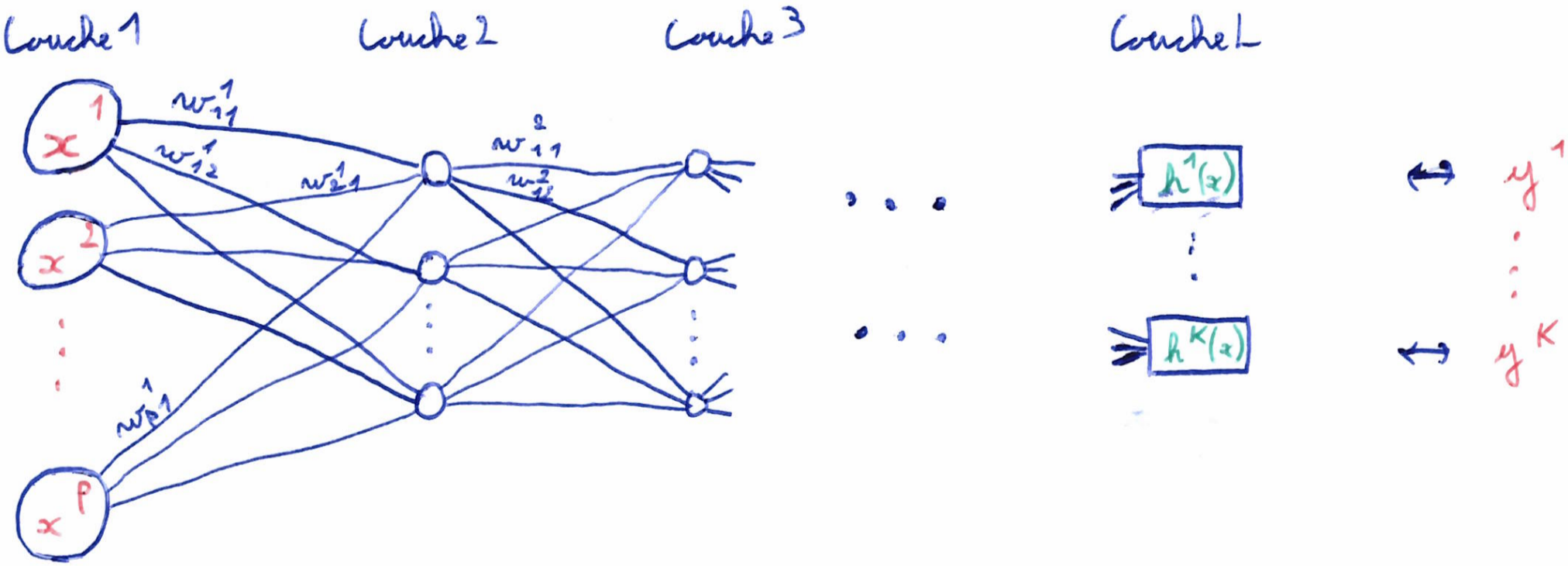
Propagation des σ_k^ℓ

Calcul de E

Retro-propagation des $\frac{\partial E}{\partial w_{pk}^\ell}$

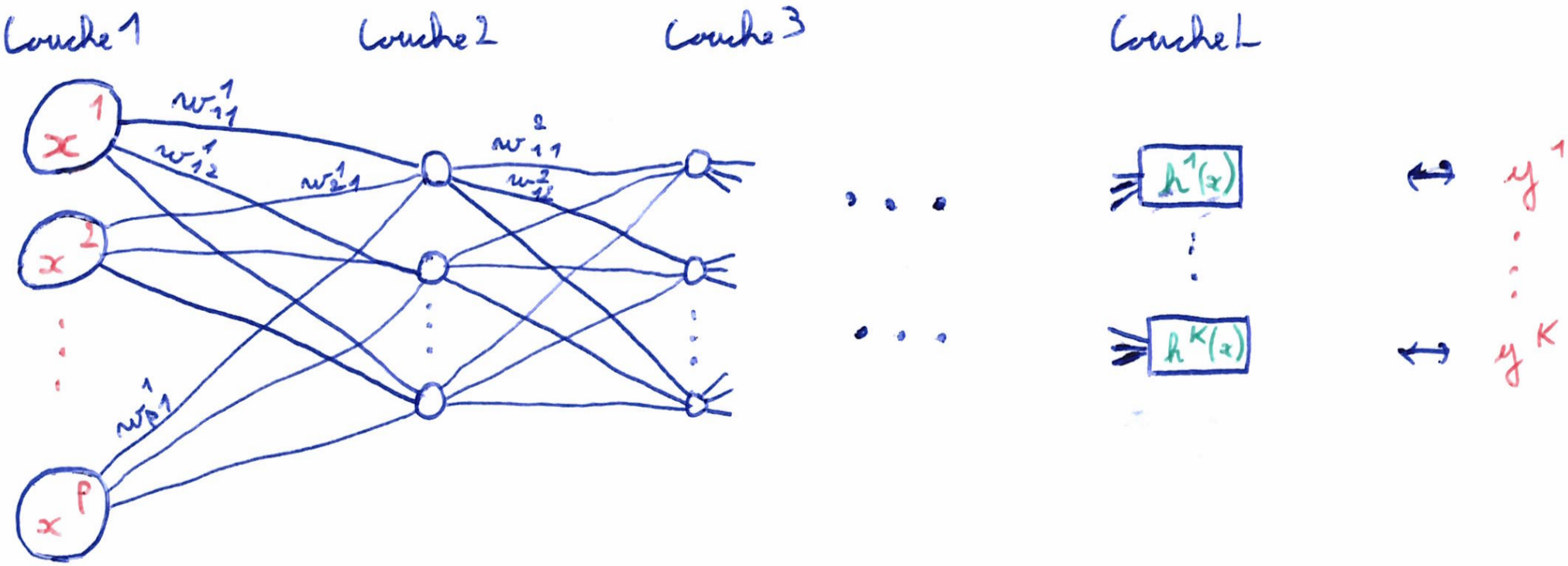
Mise à jour des poids

Calcul de $\frac{\partial E}{\partial w_{pk}^\ell}$ pour l=L-1



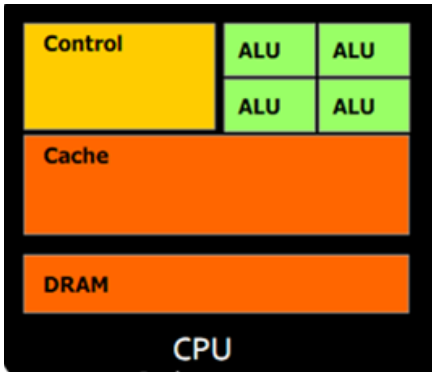
En pratique :
- Différents types de couches
- Différents types d'architecture
- Différentes variantes de descente de gradient stochastique

Dans tous les cas :
- Prédiction et apprentissage particulièrement simples à paralléliser sur GPU
- Librairie *Nvidia cuDNN* massivement utilisée et sous jacente à Caffe, Keras, TensorFlow, Theano…



Lien avec le GPU ???

```
#Create a simple model
import tensorflow as tf
x = tf.placeholder(tf.float32, [None, 784])
W = tf.Variable(tf.zeros([784, 10]))
y = tf.matmul(x, W)
```



http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU

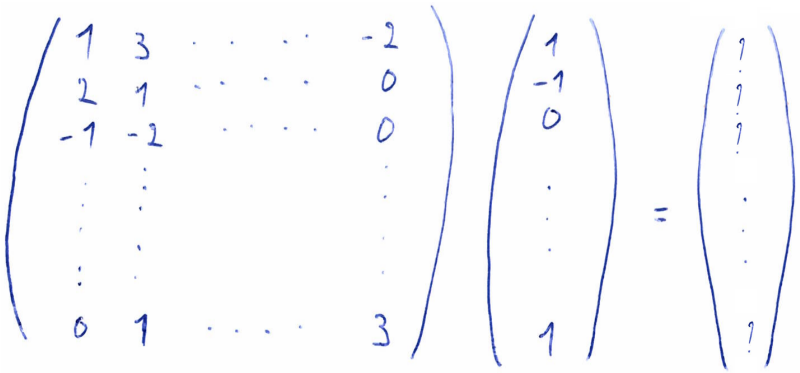
DRAM : Dynamic Random Access Memory (mémoire classique)

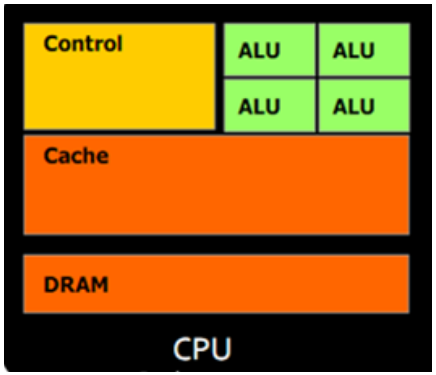
Cache : Mémoire (ici) interne au processeur. Rapide mais de taille limité

ALU : Arithmetic-Logic Unit. Effectue les calculs.

Control : Coordonne les ALU et la mémoire

Illustration : Multiplication matrice / vecteur





http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU

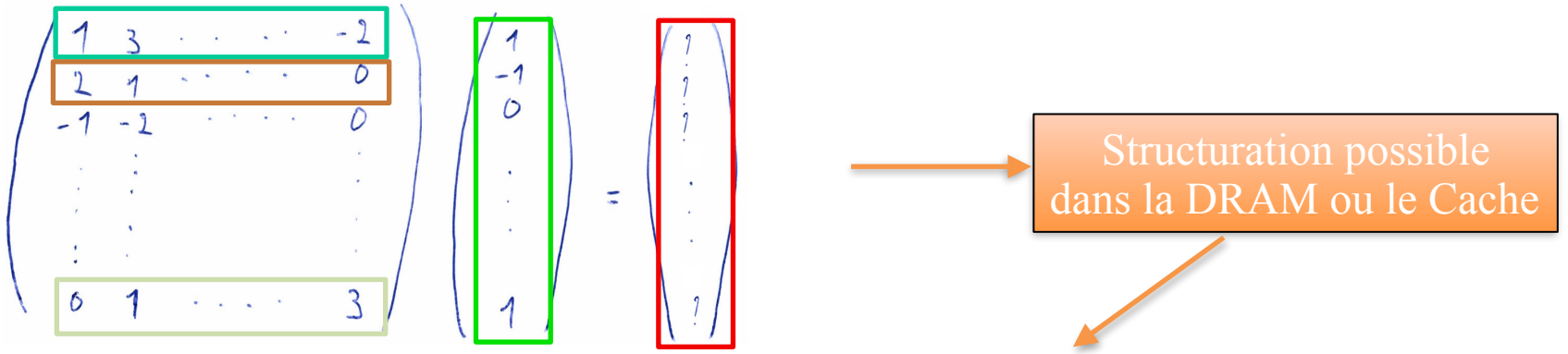
DRAM : Dynamic Random Access Memory (mémoire classique)

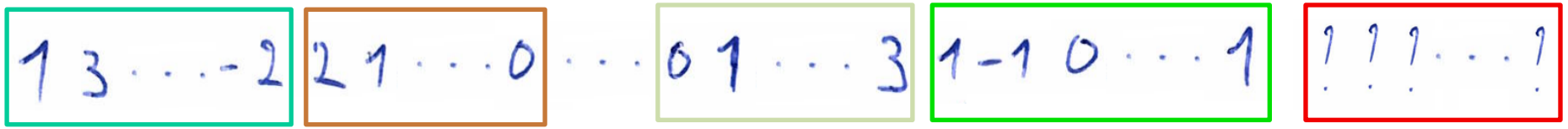
Cache : Mémoire (ici) interne au processeur. Rapide mais de taille limité

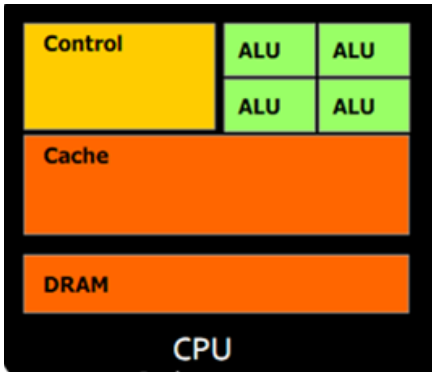
ALU : Arithmetic-Logic Unit. Effectue les calculs.

Control : Coordonne les ALU et la mémoire

Illustration : Multiplication matrice / vecteur







http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU

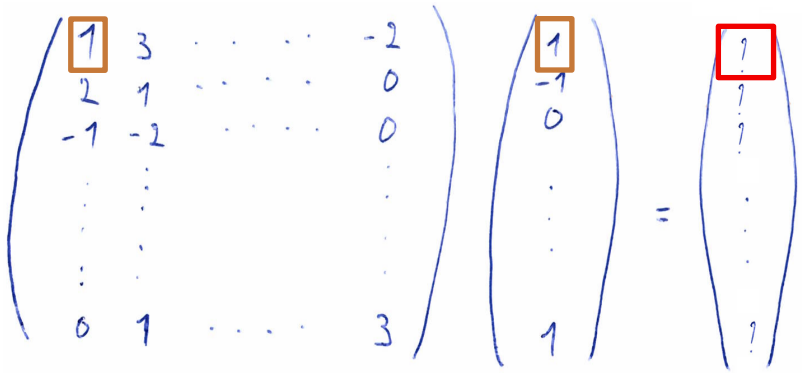
DRAM : Dynamic Random Access Memory (mémoire classique)

Cache : Mémoire (ici) interne au processeur. Rapide mais de taille limité

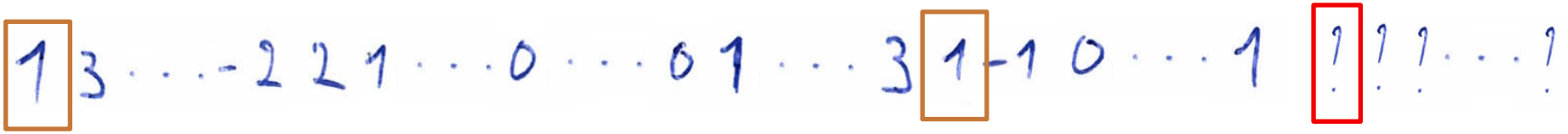
ALU : Arithmetic-Logic Unit. Effectue les calculs.

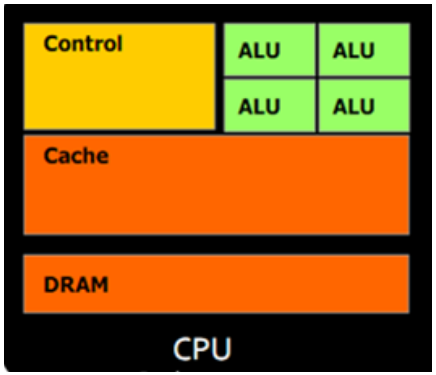
Control : Coordonne les ALU et la mémoire

Illustration : Multiplication matrice / vecteur



Opérations + et * sur différentes zones mémoires par le CPU





http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU

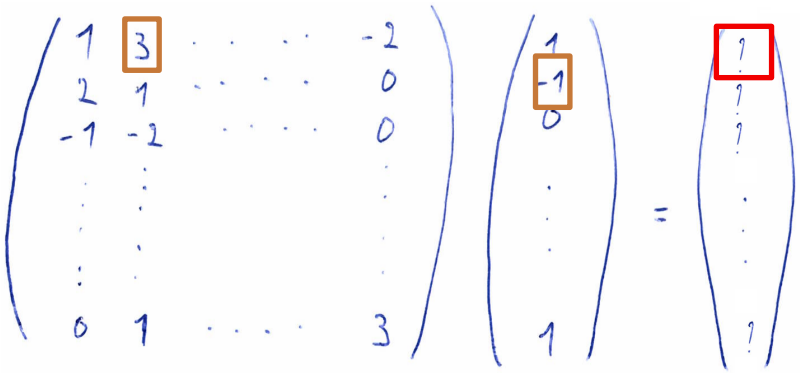
DRAM : Dynamic Random Access Memory (mémoire classique)

Cache : Mémoire (ici) interne au processeur. Rapide mais de taille limité

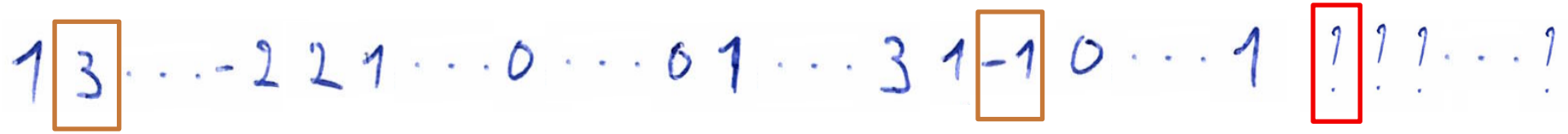
ALU : Arithmetic-Logic Unit. Effectue les calculs.

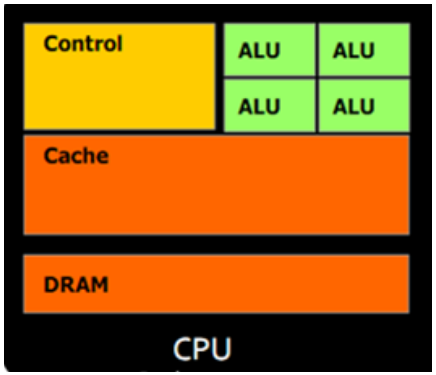
Control : Coordonne les ALU et la mémoire

Illustration : Multiplication matrice / vecteur



Opérations + et * sur différentes zones mémoires par le CPU





http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU

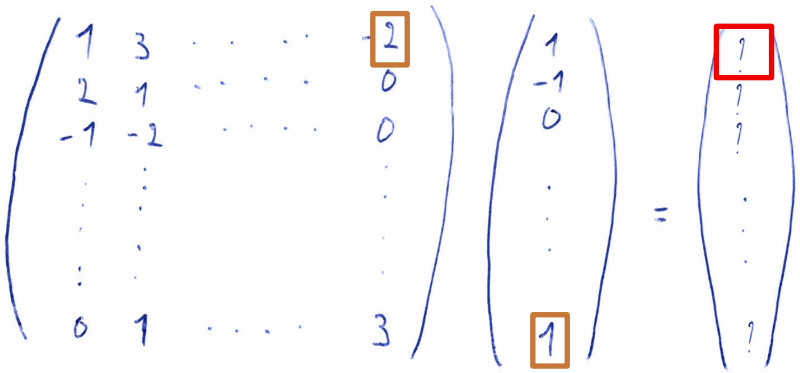
DRAM : Dynamic Random Access Memory (mémoire classique)

Cache : Mémoire (ici) interne au processeur. Rapide mais de taille limité

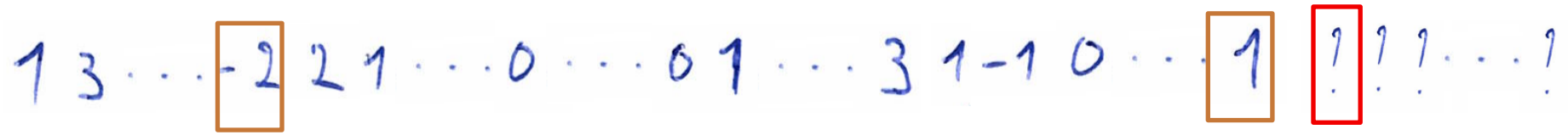
ALU : Arithmetic-Logic Unit. Effectue les calculs.

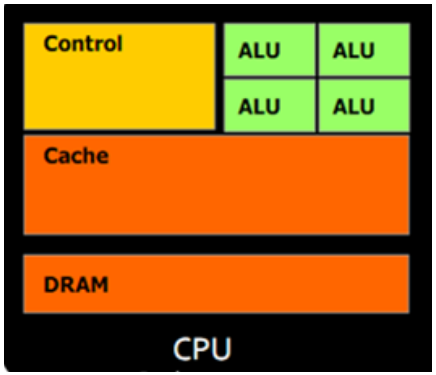
Control : Coordonne les ALU et la mémoire

Illustration : Multiplication matrice / vecteur



Opérations + et * sur différentes zones mémoires par le CPU





http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU

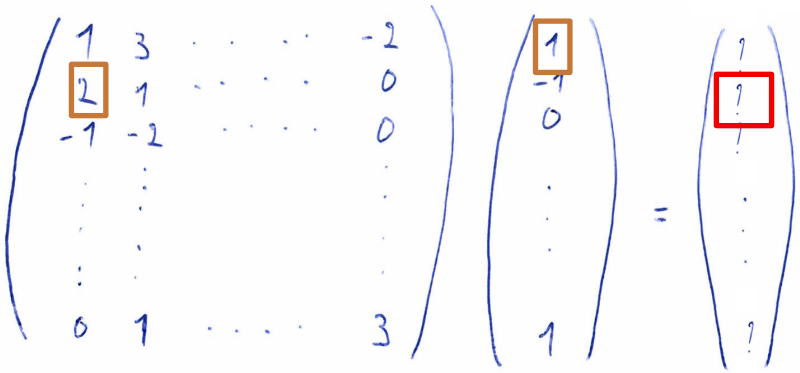
DRAM : Dynamic Random Access Memory (mémoire classique)

Cache : Mémoire (ici) interne au processeur. Rapide mais de taille limité

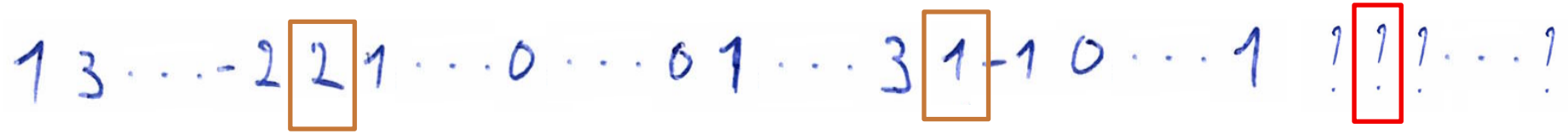
ALU : Arithmetic-Logic Unit. Effectue les calculs.

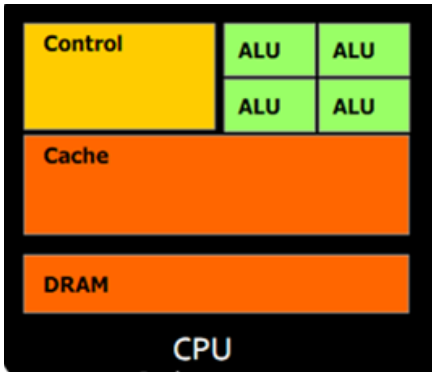
Control : Coordonne les ALU et la mémoire

Illustration : Multiplication matrice / vecteur



Opérations + et * sur différentes zones mémoires par le CPU





http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU

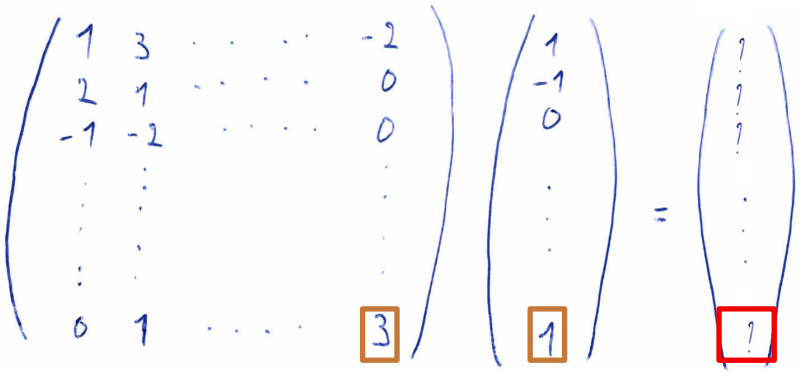
DRAM : Dynamic Random Access Memory (mémoire classique)

Cache : Mémoire (ici) interne au processeur. Rapide mais de taille limité

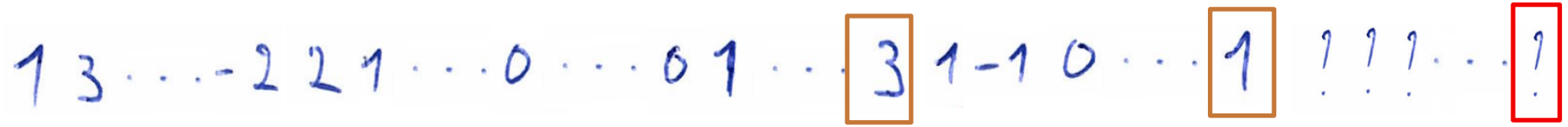
ALU : Arithmetic-Logic Unit. Effectue les calculs.

Control : Coordonne les ALU et la mémoire

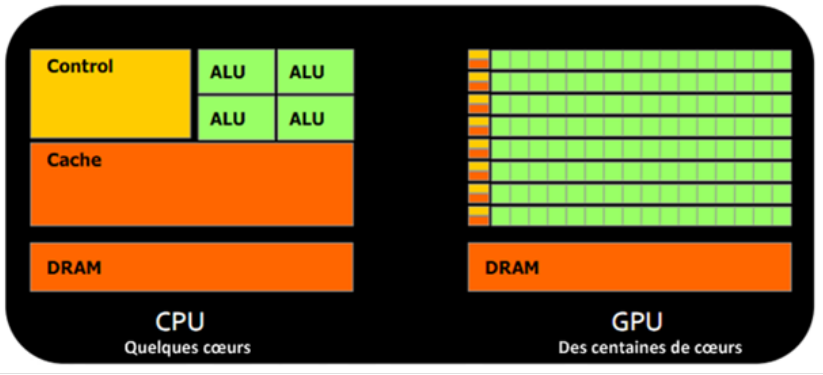
Illustration : Multiplication matrice / vecteur



Opérations + et * sur différentes zones mémoires par le CPU



6.c) Apprentissage machine et GPU — GPU



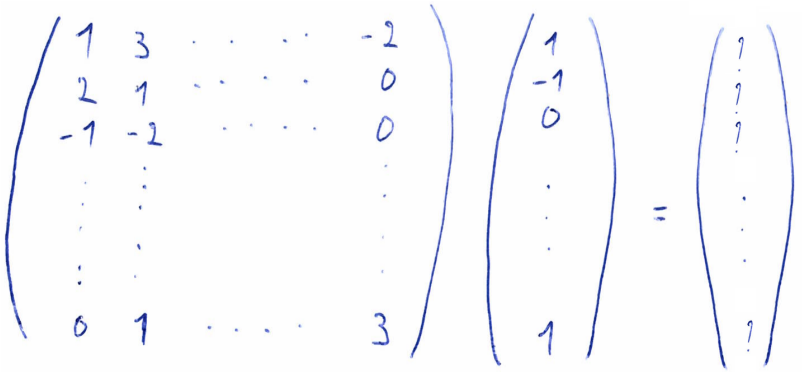
http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU

DRAM : Dynamic Random Access Memory (mémoire classique)

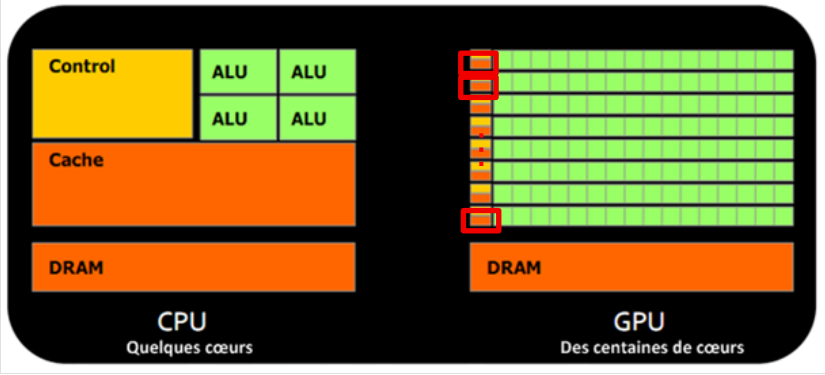
Cache : Mémoire (ici) interne au processeur. Rapide mais de taille limité

ALU : Arithmetic-Logic Unit. Effectue les calculs.

Control : Coordonne les ALU et la mémoire



6.c) Apprentissage machine et GPU — GPU



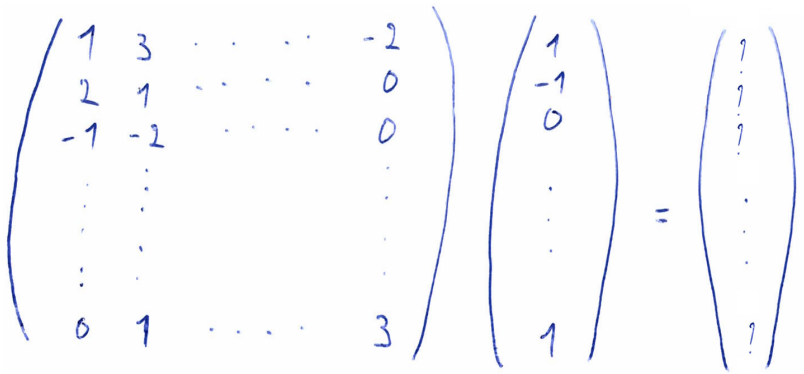
http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU

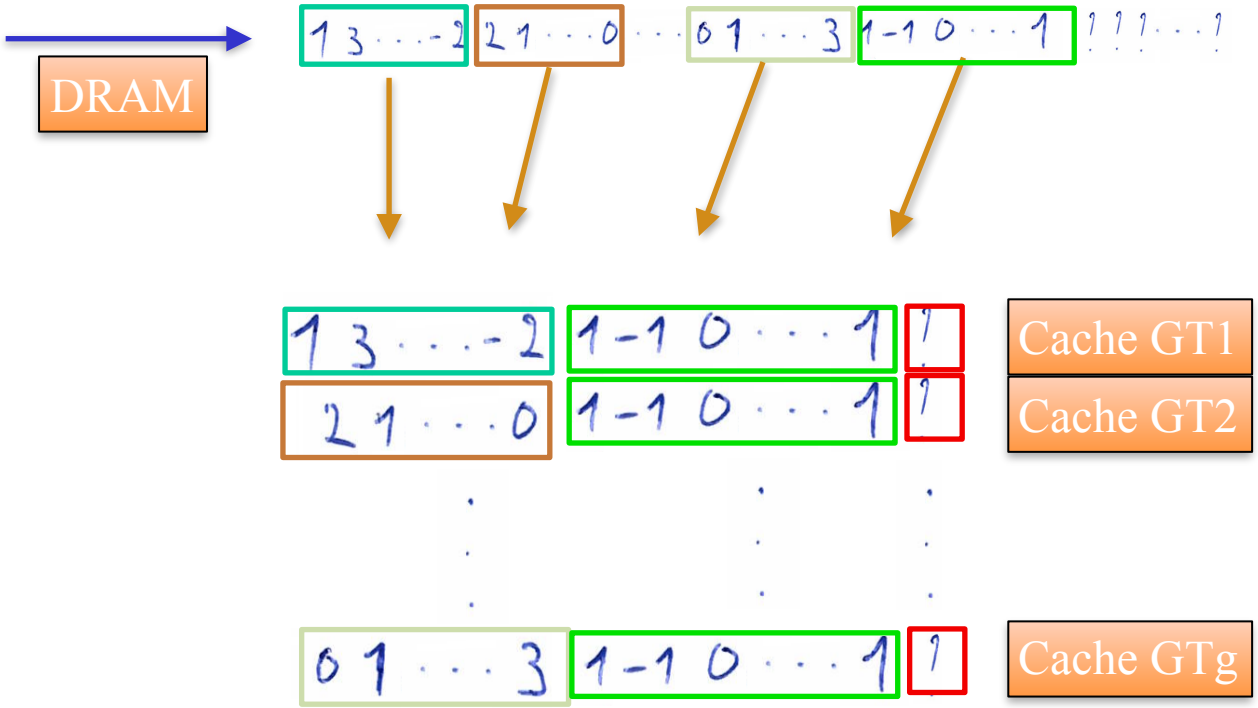
DRAM : Dynamic Random Access Memory (mémoire classique)

Cache : Mémoire (ici) interne au processeur. Rapide mais de taille limité

ALU : Arithmetic-Logic Unit. Effectue les calculs.

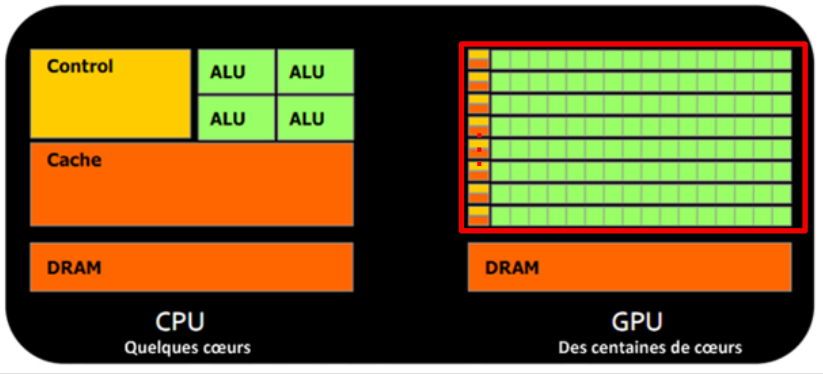
Control : Coordonne les ALU et la mémoire





Etape 1 : copies des données dans les différents groupes de travail et allocation mémoire (plus gestion de flux)

6.c) Apprentissage machine et GPU — GPU



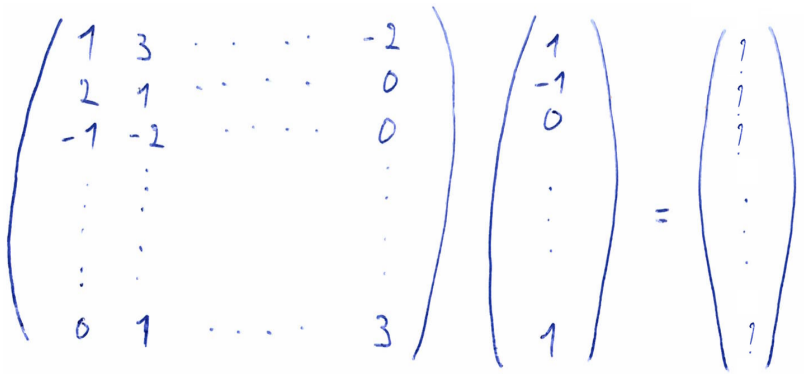
http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU

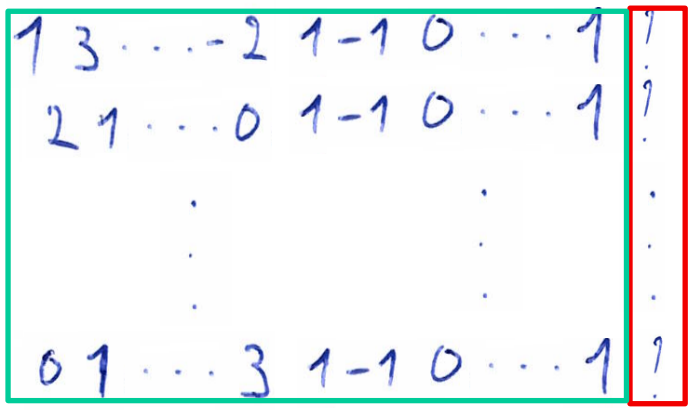
DRAM : Dynamic Random Access Memory (mémoire classique)

Cache : Mémoire (ici) interne au processeur. Rapide mais de taille limité

ALU : Arithmetic-Logic Unit. Effectue les calculs.

Control : Coordonne les ALU et la mémoire



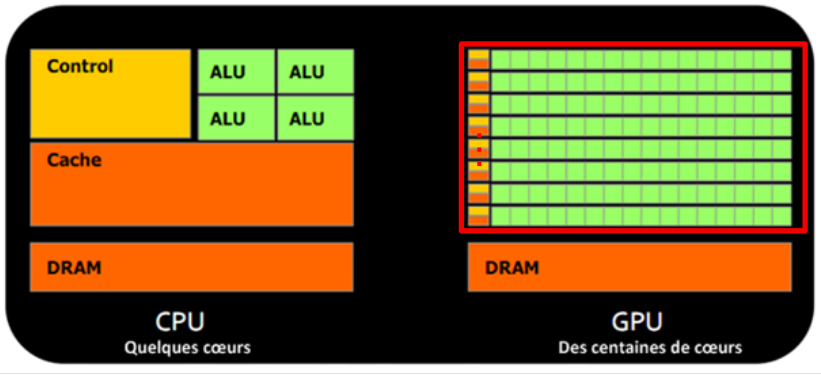


Cache GT1

Cache GT2

Cache GTg

Etape 2 : Toutes les multiplications et additions sont faites en parallèle



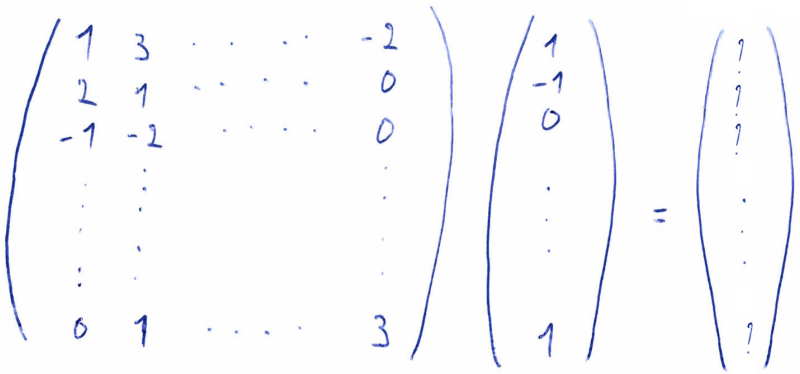
http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU

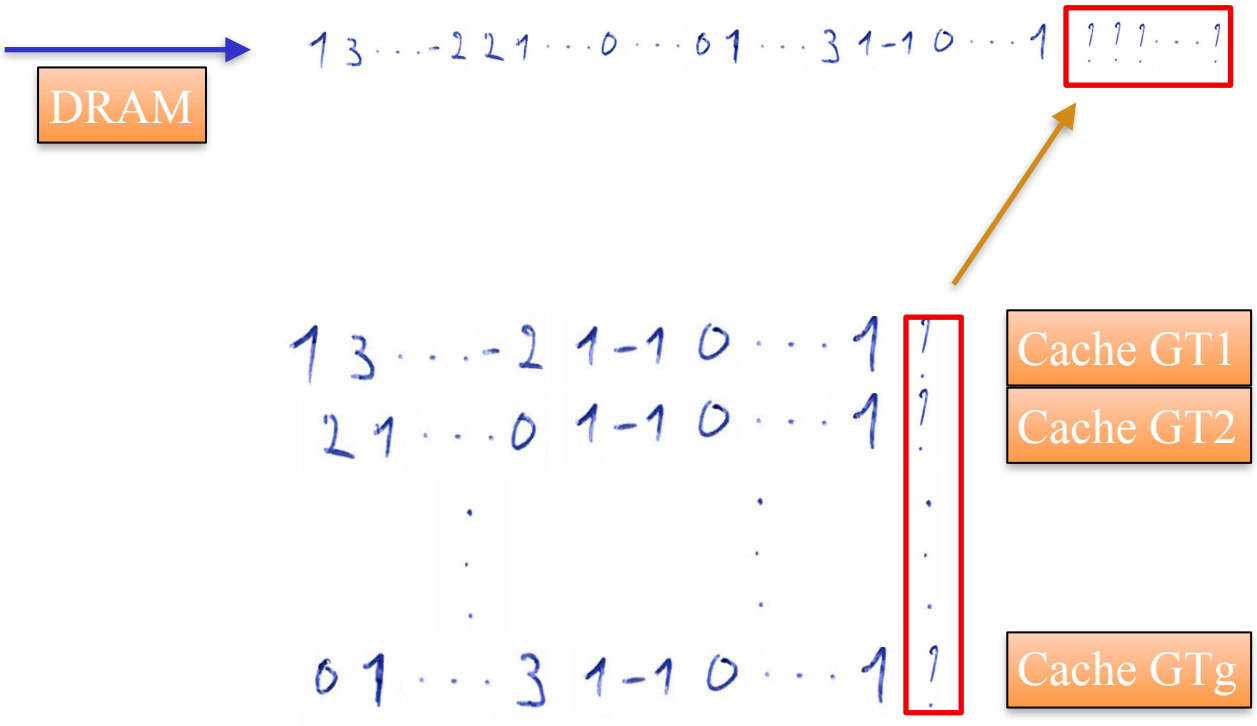
DRAM : Dynamic Random Access Memory (mémoire classique)

Cache : Mémoire (ici) interne au processeur. Rapide mais de taille limité

ALU : Arithmetic-Logic Unit. Effectue les calculs.

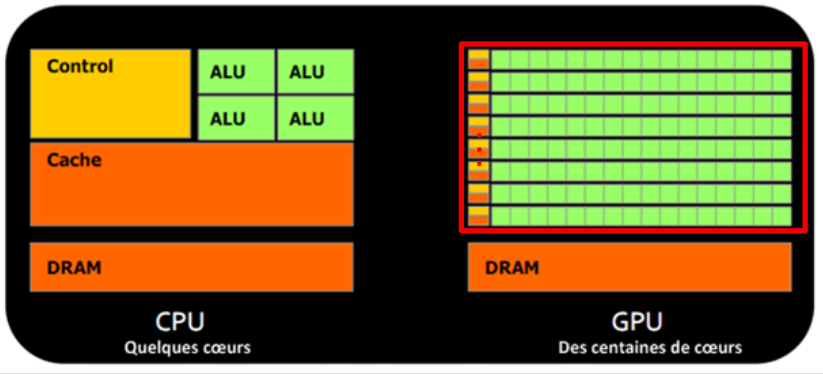
Control : Coordonne les ALU et la mémoire





Etape 3 : Résultat copié dans la DRAM

6.c) Apprentissage machine et GPU — GPU



http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU

DRAM : Dynamic Random Access Memory (mémoire classique)

Cache : Mémoire (ici) interne au processeur. Rapide mais de taille limité

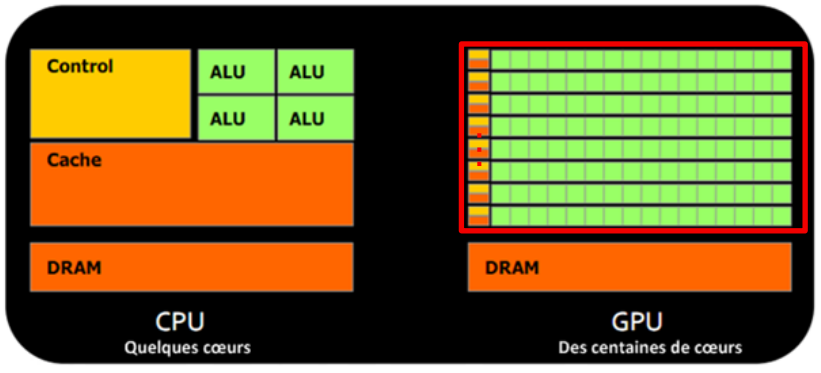
ALU : Arithmetic-Logic Unit. Effectue les calculs.

Control : Coordonne les ALU et la mémoire

Réflexions pour la programmation bas-niveau :
- [Gain de temps en parallélisant les calculs] - [perte de temps dû aux transferts de données] *> 0 ?*
- Peut-on rendre négligeable les temps de transferts (latence) ?

Réflexions pour la programmation haut-niveau :
- Est-ce que les mêmes instructions (additions, multiplications, …) vont s'effectuer sur de gros blocs de données ?
- Est-ce que ces blocs de données sont continus en mémoire ?

6.c) Apprentissage machine et GPU — GPU



http://igm.univ-mlv.fr/~dr/XPOSE2013/GPGPU

DRAM : Dynamic Random Access Memory (mémoire classique)

Cache : Mémoire (ici) interne au processeur. Rapide mais de taille limité

ALU : Arithmetic-Logic Unit. Effectue les calculs.

Control : Coordonne les ALU et la mémoire

Réflexions pour la programmation bas-niveau :
- [Gain de temps en parallélisant les calculs] - [perte de temps dû aux transferts de données] *> 0 ?*
- Peut-on rendre négligeable les temps de transferts (latence) ?

Réflexions pour la programmation haut-niveau :
- Est-ce que les mêmes instructions (additions, multiplications, …) vont s'effectuer sur de gros blocs de données ?
- Est-ce que ces blocs de données sont continus en mémoire ?

Intérêt sur d'autres algorithmes de machine learning :
- Succes de XGboost, librairie Nvidia Rapids, …
- Intérêt sur des graphes ou d'autres données peu régulières en espace ???

7) Conclusion

- Champ de recherche très actif et applications nombreuses
- Formalisme de description des données bien défini
- Lien avec le web des données ?

MERCI !!!