Régressions polynomiales

Cours 2 – Régularisation et sélection de modèle

1. Évaluation d'un modèle

1.1 Généralisation et surapprentissage

Défi principal de l'apprentissage supervisé :

– Il est relativement facile d'entraîner un modèle qui « marche » bien (faible erreur de prédiction) sur les données d'apprentissage
Exemple extrême : apprentissage « par cœur »

1.1 Généralisation et surapprentissage

Défi principal de l'apprentissage supervisé :

- Il est relativement facile d'entraîner un modèle qui « marche » bien (faible erreur de prédiction) sur les données d'apprentissage
  Exemple extrême : apprentissage « par cœur »
- **Généralisation :** capacité du modèle à **faire de bonnes prédictions sur des données dont on ne connaît pas l'étiquette**

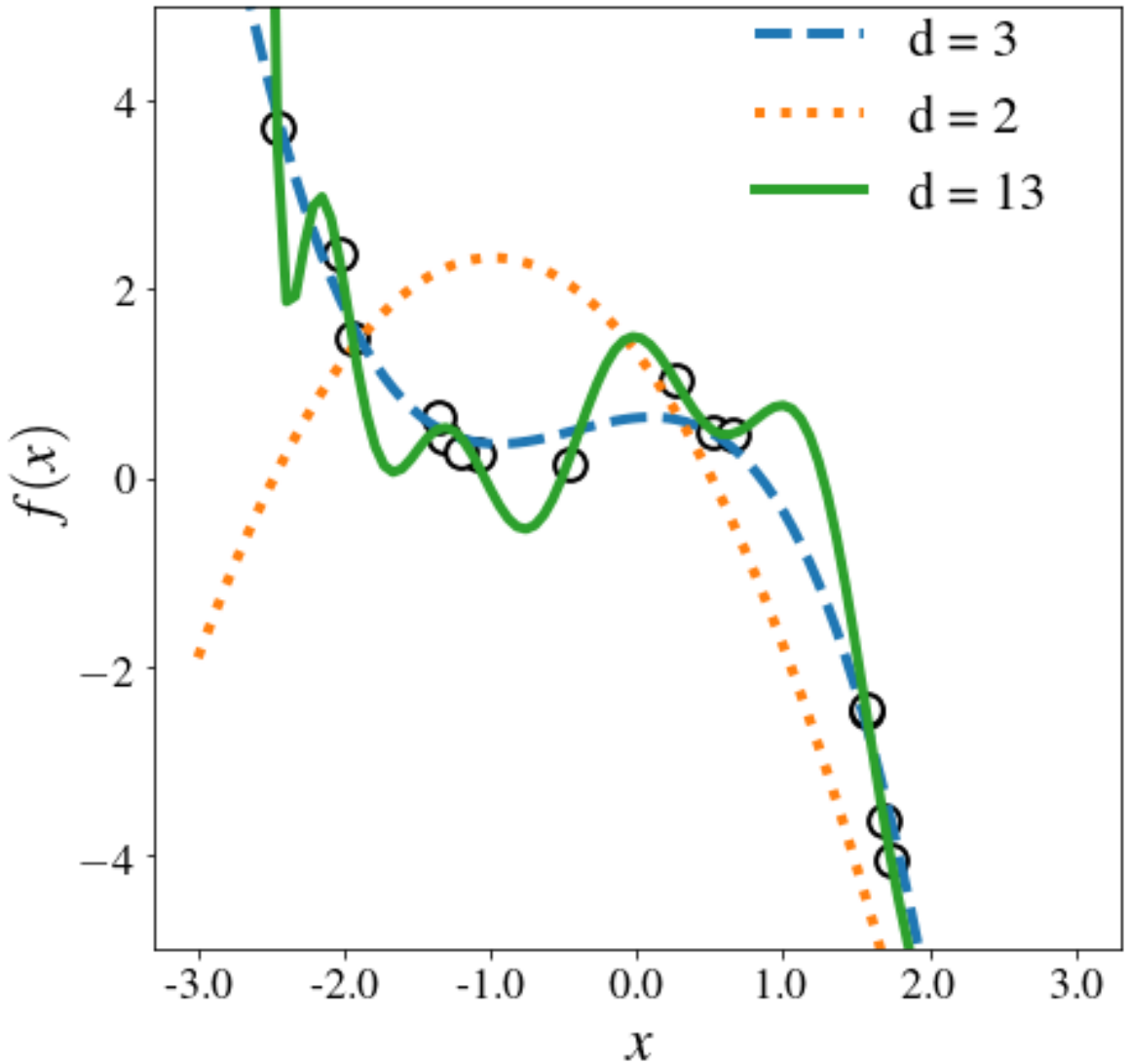
1.1 Généralisation et surapprentissage

Défi principal de l'apprentissage supervisé :

- Il est relativement facile d'entraîner un modèle qui « marche » bien (faible erreur de prédiction) sur les données d'apprentissage
  Exemple extrême : apprentissage « par cœur »
- **Généralisation :** capacité du modèle à **faire de bonnes prédictions sur des données dont on ne connaît pas l'étiquette**
- **Surapprentissage :** quand la performance est meilleure sur les données d'apprentissage que sur de nouvelles données (overfitting)

Surapprentissage

Simulation : vrai modèle = polynôme de degré 3 + bruit.



Compromis biais-variance

1.2 Jeux d'entraînement et de test

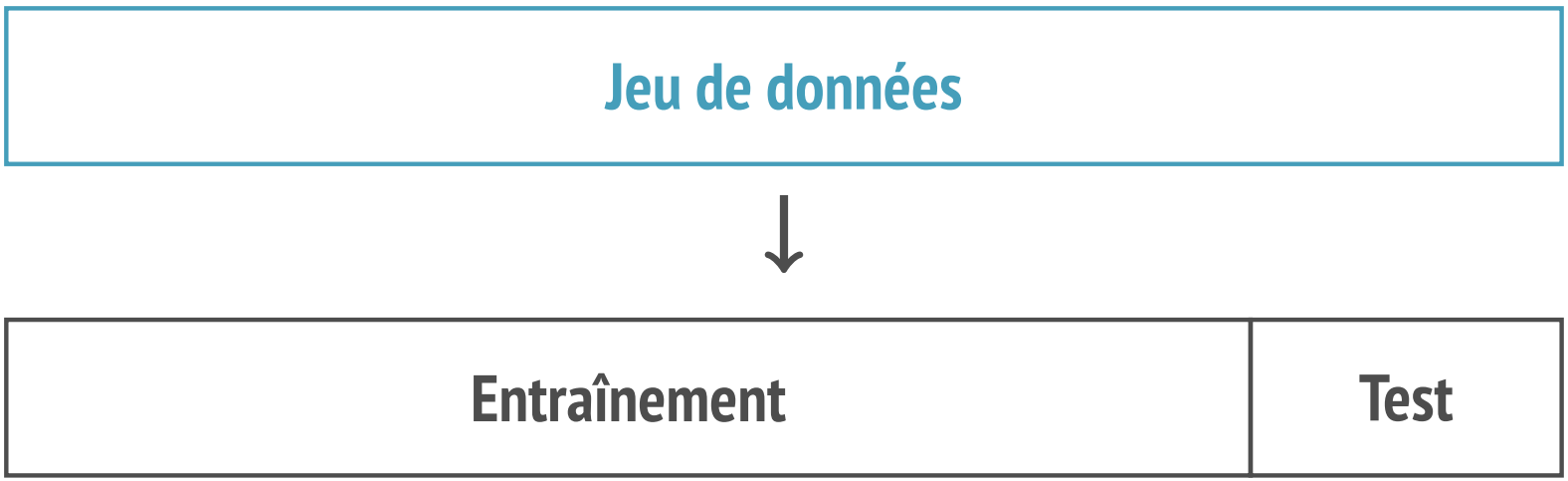
- **Erreur de généralisation :** erreur que l'on peut attendre sur de nouvelles données (la définition formelle fait appel à l'espérance)

1.2 Jeux d'entraînement et de test

- **Erreur de généralisation :** erreur que l'on peut attendre sur de nouvelles données (la définition formelle fait appel à l'espérance)
- Elle est **estimée** en **mettant de côté** une partie des données :
  - On sépare les données en un **jeu d'entraînement** et un **jeu de test** (typiquement, 80%–20%)

1.2 Jeux d'entraînement et de test

- **Erreur de généralisation :** erreur que l'on peut attendre sur de nouvelles données (la définition formelle fait appel à l'espérance)
- Elle est **estimée** en **mettant de côté** une partie des données :
  - On sépare les données en un **jeu d'entraînement** et un **jeu de test** (typiquement, 80%–20%)



  - Le **jeu d'entraînement** (train set) sert à apprendre le modèle
  - Le **jeu de test** sert à estimer l'erreur de généralisation du modèle

Règle d'or

NE PAS TOUCHER au jeu de test

sauf pour évaluer l'erreur de généralisation du modèle

1.3 Validation croisée

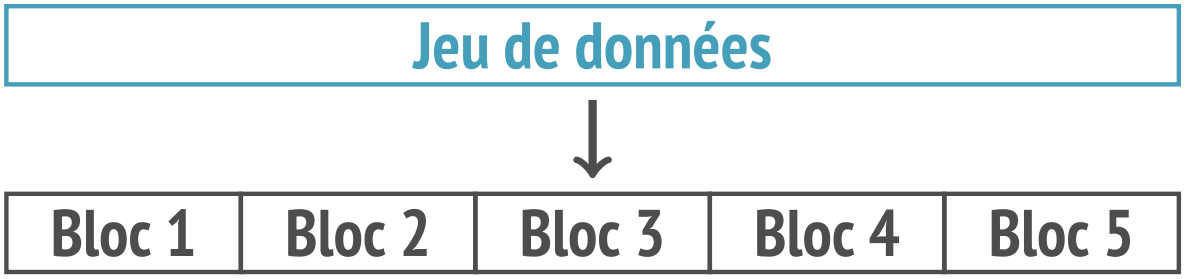
- La **validation croisée** (cross-validation) permet :
  - d'utiliser toutes les données pour l'entraînement et pour la validation
  - d'obtenir une performance moyenne (+/- écart-type) moins sensible au choix du jeu de test
- On sépare le jeu de données en K **blocs** (folds)

  En pratique, K=5 ou K=10 le plus souvent (équilibre entre le nombre d'expériences et la taille de chaque jeu d'entraînement)
- On utilise tour à tour chacun des blocs comme **jeu de validation** et l'union des autres comme **jeu d'entraînement**

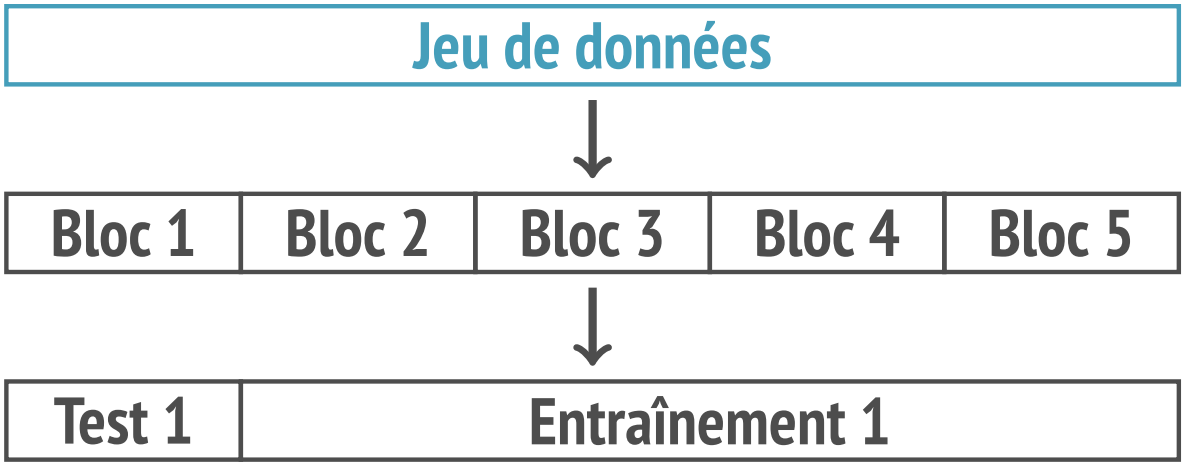
$\Rightarrow$ K scores de performance $\rightarrow$ **performance de généralisation** du modèle

1.3 Validation croisée

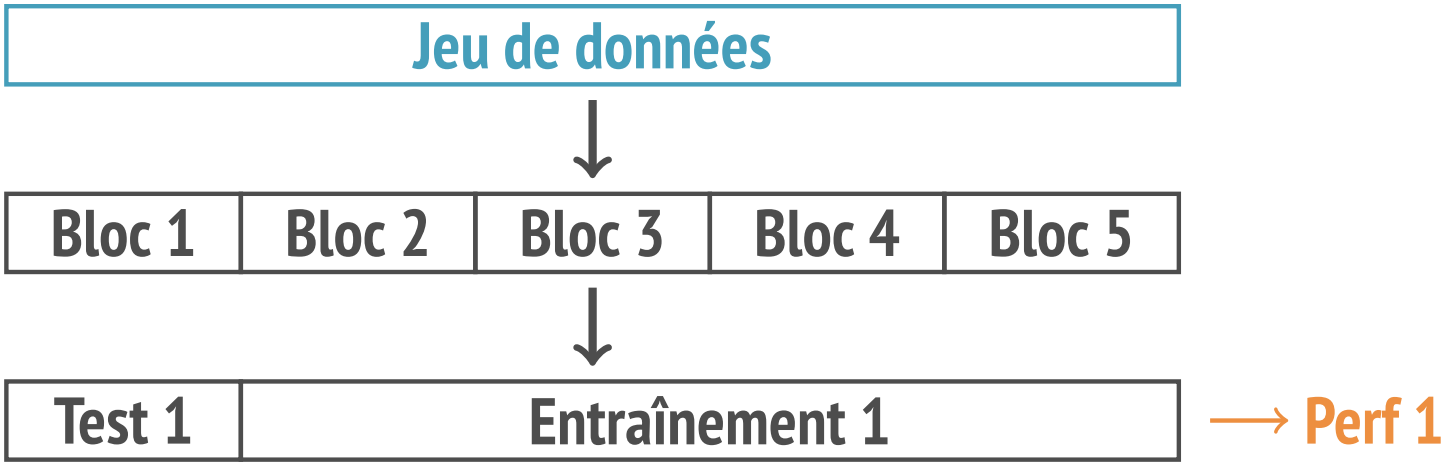
1.3 Validation croisée



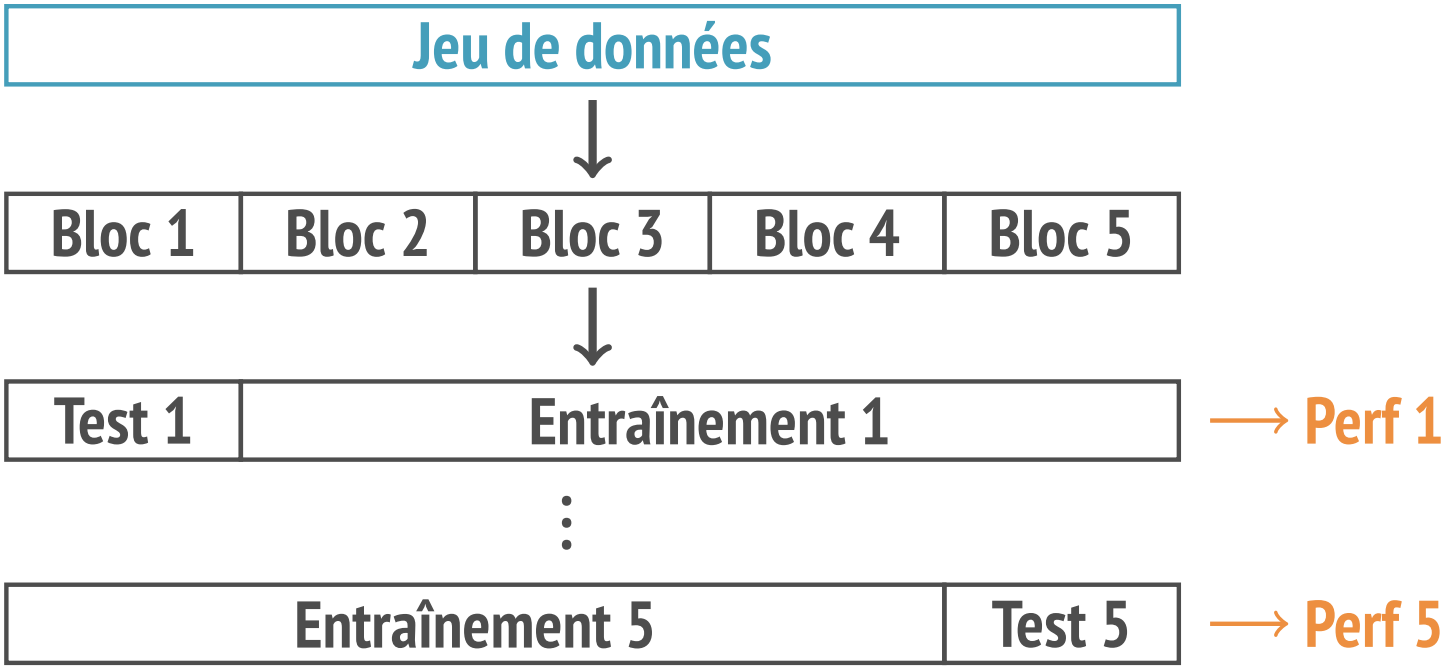
1.3 Validation croisée



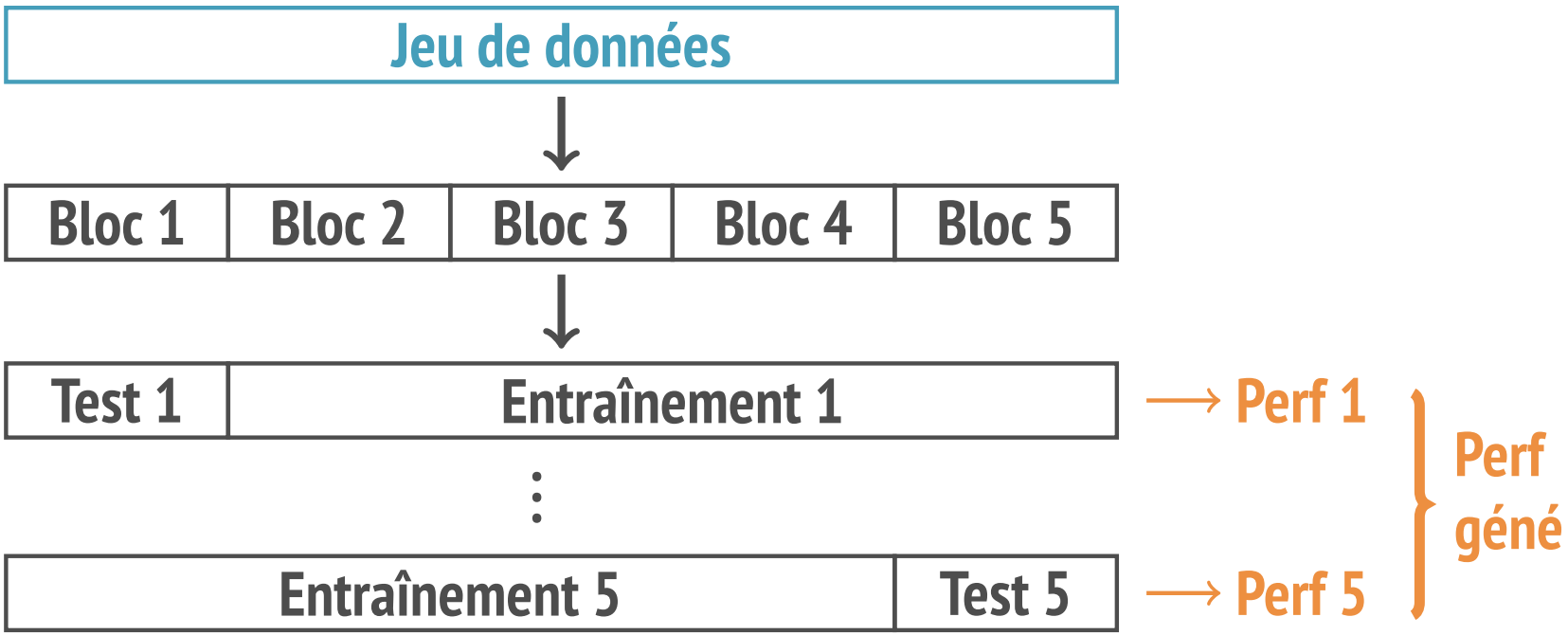
1.3 Validation croisée



1.3 Validation croisée



1.3 Validation croisée



1.4 Évaluation d'un modèle (classification)

- **Pourcentage d'erreur :** proportion d'observations mal classifiées

1.4 Évaluation d'un modèle (classification)

- **Pourcentage d'erreur :** proportion d'observations mal classifiées
- **Problème :** cas où les classes ne sont **pas équilibrées**

  Exemple : détection de fraude
  - 99% des observations ne sont pas des fraudes
  - Un modèle qui prédit toujours « non » a un pourcentage d'erreur de 1%.

1.4 Évaluation d'un modèle (classification)

- **Pourcentage d'erreur :** proportion d'observations mal classifiées
- **Problème :** cas où les classes ne sont **pas équilibrées**

  Exemple : détection de fraude
  - 99% des observations ne sont pas des fraudes
  - Un modèle qui prédit toujours « non » a un pourcentage d'erreur de 1%.
- **Matrice de confusion** (confusion matrix)

		Classe réelle	
		0	1
Classe prédite	0	Vrais Négatifs (TN)	Faux Négatifs (FN)
	1	Faux Positifs (FP)	Vrais Positifs (TP)

1.4 Évaluation d'un modèle (classification)

- **Pourcentage d'erreur :** proportion d'observations mal classifiées
- **Problème :** cas où les classes ne sont **pas équilibrées**

  Exemple : détection de fraude
  - 99% des observations ne sont pas des fraudes
  - Un modèle qui prédit toujours « non » a un pourcentage d'erreur de 1%.
- **Matrice de confusion** (confusion matrix)

		True class	
		0	1
Predicted class	0	True Negatives (TN)	False Negatives (FN)
	1	False Positives (FP)	True Positives (TP)

1.3 Évaluation d'un modèle (classification)

		Classe réelle		
		0	1	
Classe prédite	0	Vrais Négatifs (TN)	Faux Négatifs (FN)	TN+FN
	1	Faux Positifs (FP)	Vrais Positifs (TP)	TP+FP
		TN + FP	TP + FN	

- Sensibilité (sensitivity) = rappel (recall) = taux de vrais positifs = $\frac{TP}{TP + FN}$
- Spécificité (specificity) = taux de vrais négatifs (TN rate) = $\frac{TN}{TN + FP}$
- Précision (precision) = $\frac{TP}{TP + FP}$
- Précision (accuracy) = $\frac{TP + TN}{TP + FP + TN + FN}$
- Score F (F-score) = moyenne harmonique (précision, rappel) = $\frac{2\,TP}{TP + FP + TP + FN}$

1.5 Évaluation d'un modèle (régression)

- **Compter** le nombre d'erreurs n'a pas de sens

1.5 Évaluation d'un modèle (régression)

- **Compter** le nombre d'erreurs n'a pas de sens
- Somme des carrés des erreurs : RSS $= \sum_{i=1}^{n} (y_i - f(\vec{x}_i))^2$
- Racine de l'erreur quadratique moyenne (Root Mean Squared Error) :

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - f(\vec{x}_i))^2}$$

- Erreur carrée relative : RSE $= \frac{\sum_{i=1}^{n} (y_i - f(\vec{x}_i))^2}{\sum_{i=1}^{n} (y_i - \overline{y})^2}$ $\quad \overline{y} = \sum_{l=1}^{n} y_l$
- Coefficient de détermination :

$$R^2 = 1 - \text{RSE} = \frac{\sum_{i=1}^{n} (y_i - \overline{y}) \left(f(\vec{x}_i) - \overline{f(\vec{x})} \right)}{\sqrt{\sum_{i=1}^{n} (y_i - \overline{y})^2} \sqrt{\sum_{i=1}^{n} \left(f(\vec{x}_i) - \overline{f(\vec{x})} \right)^2}}$$

1.6 Sélection de modèle

- Comment déterminer **le meilleur modèle** parmi ceux appris :
  - avec différents algorithmes d'apprentissage ;
  - avec différentes valeurs d'hyperparamètre(s) pour le même algorithme ?

1.6 Sélection de modèle

- Comment déterminer **le meilleur modèle** parmi ceux appris :
  - avec différents algorithmes d'apprentissage ;
  - avec différentes valeurs d'hyperparamètre(s) pour le même algorithme ?
- **Idée :** sélectionner celui qui a la **meilleure performance sur le jeu de test.**

1.6 Sélection de modèle

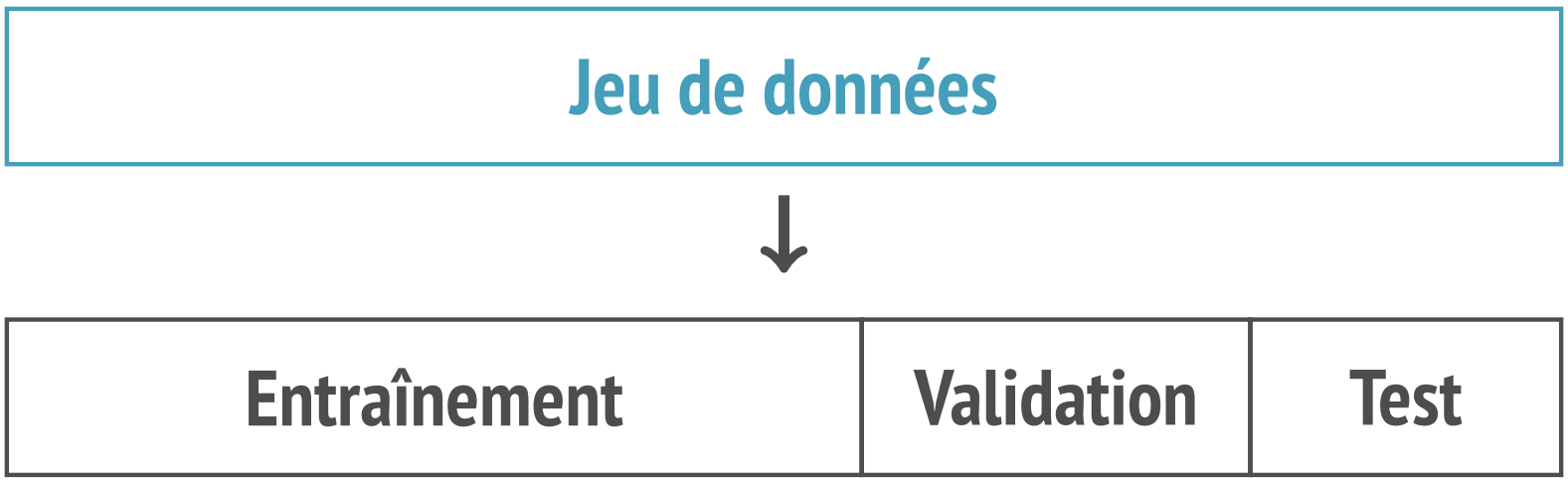
- Comment déterminer **le meilleur modèle** parmi ceux appris :
  - avec différents algorithmes d'apprentissage ;
  - avec différentes valeurs d'hyperparamètre(s) pour le même algorithme ?
- **Idée :** sélectionner celui qui a la **meilleure performance sur le jeu de test.**
- **Problème :** on ne peut plus déterminer l'**erreur de généralisation** car les données de test ont déjà servi.

1.6 Sélection de modèle

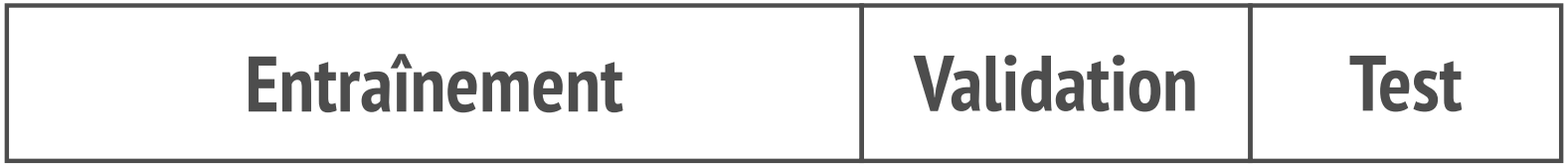
- Comment déterminer **le meilleur modèle** parmi ceux appris :
  - avec différents algorithmes d'apprentissage ;
  - avec différentes valeurs d'hyperparamètre(s) pour le même algorithme ?
- **Idée :** sélectionner celui qui a la **meilleure performance sur le jeu de test.**
- **Problème :** on ne peut plus déterminer l'**erreur de généralisation** car les données de test ont déjà servi.
- On sépare les données en 3 jeux : apprentissage, **validation** et test.

1.6 Sélection de modèle

- Comment déterminer **le meilleur modèle** parmi ceux appris :
  - avec différents algorithmes d'apprentissage ;
  - avec différentes valeurs d'hyperparamètre(s) pour le même algorithme ?
- **Idée :** sélectionner celui qui a la **meilleure performance sur le jeu de test.**
- **Problème :** on ne peut plus déterminer l'**erreur de généralisation** car les données de test ont déjà servi.
- On sépare les données en 3 jeux : apprentissage, **validation** et test.



Jeu de validation



- Le **jeu d'entraînement** sert à apprendre **chacun des** modèles
- Le **jeu de validation** sert à **sélectionner** le meilleur modèle
- Le **jeu de test** sert à estimer l'erreur de généralisation **du modèle sélectionné**

Jeu de validation

Entraînement	Validation	Test

- Le **jeu d'entraînement** sert à apprendre **chacun des** modèles
- Le **jeu de validation** sert à **sélectionner** le meilleur modèle
- Le **jeu de test** sert à estimer l'erreur de généralisation **du modèle sélectionné**
- **Problèmes :**
  - On utilise seulement une fraction des données pour l'entraînement
  - Possiblité que le modèle sélectionné ait une meilleure performance sur ce jeu de validation précis à cause d'un artefact.

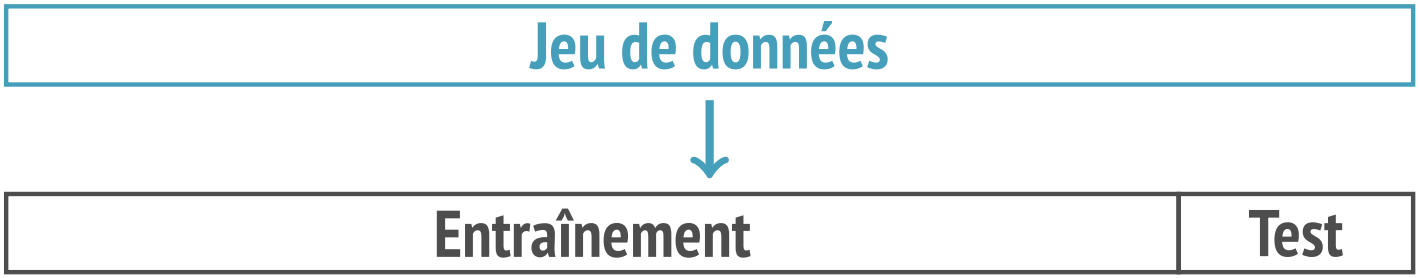
Validation croisée (sélection de modèle)

- On peut aussi utiliser la **validation croisée** pour la **sélection de modèle**
- On sépare le jeu d'entraînement en K **blocs** (ou folds)
- Pour chaque algorithme ou valeur d'hyperparamètre à évaluer :
  - On utilise tour à tour chacun des blocs comme **jeu de validation** et l'union des autres comme **jeu d'entraînement**

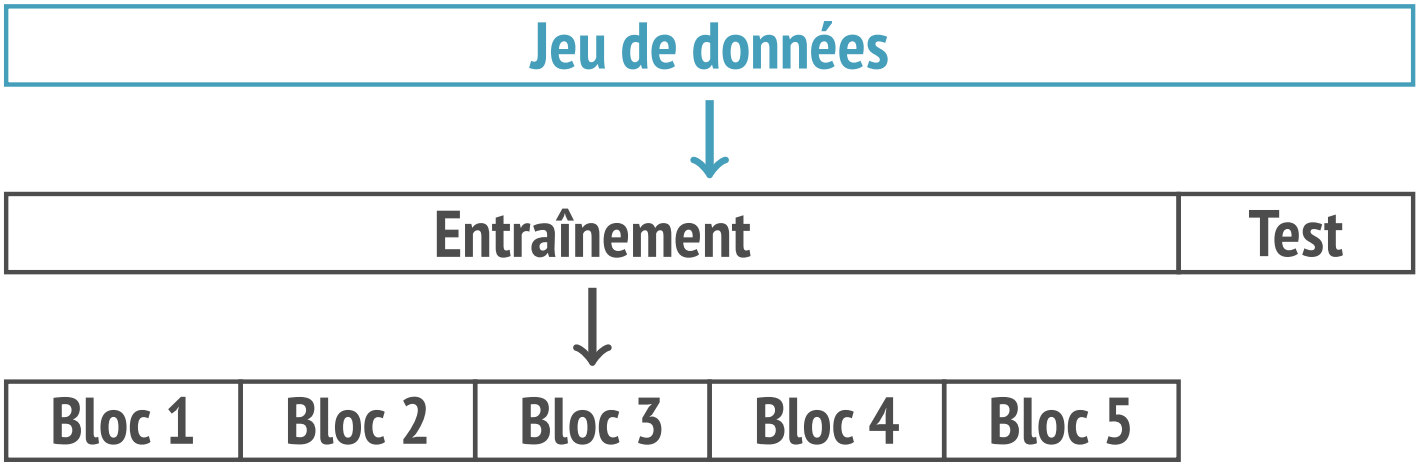
  $\Rightarrow$ on obtient K scores de performance $\rightarrow$ **performance moyenne**
- On sélectionne l'algorithme ou la valeur d'hyperparamètre qui donne la meilleure performance moyenne
- On réentraîne sur le jeu d'entraînement l'algorithme sélectionné
- On estime l'erreur de généralisation en évaluant sur le jeu de test le modèle ainsi obtenu

Validation croisée (sélection de modèle)

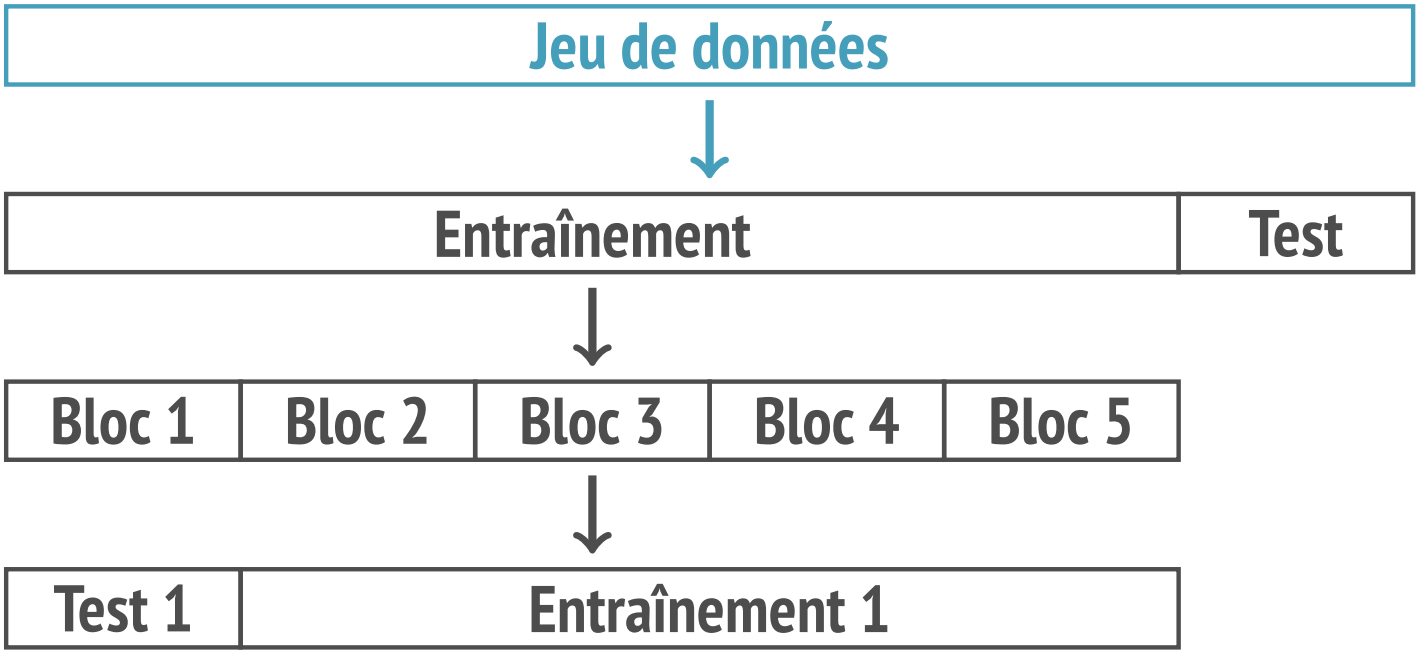
Validation croisée (sélection de modèle)



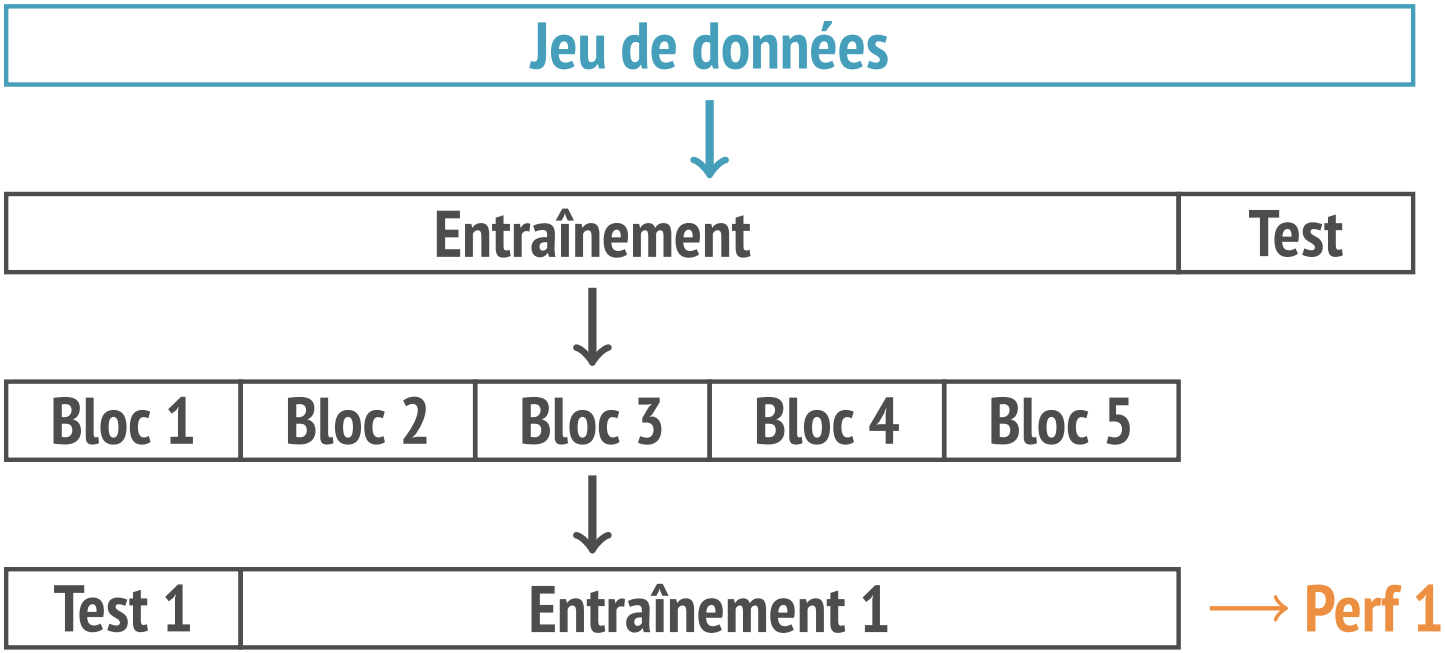
Validation croisée (sélection de modèle)



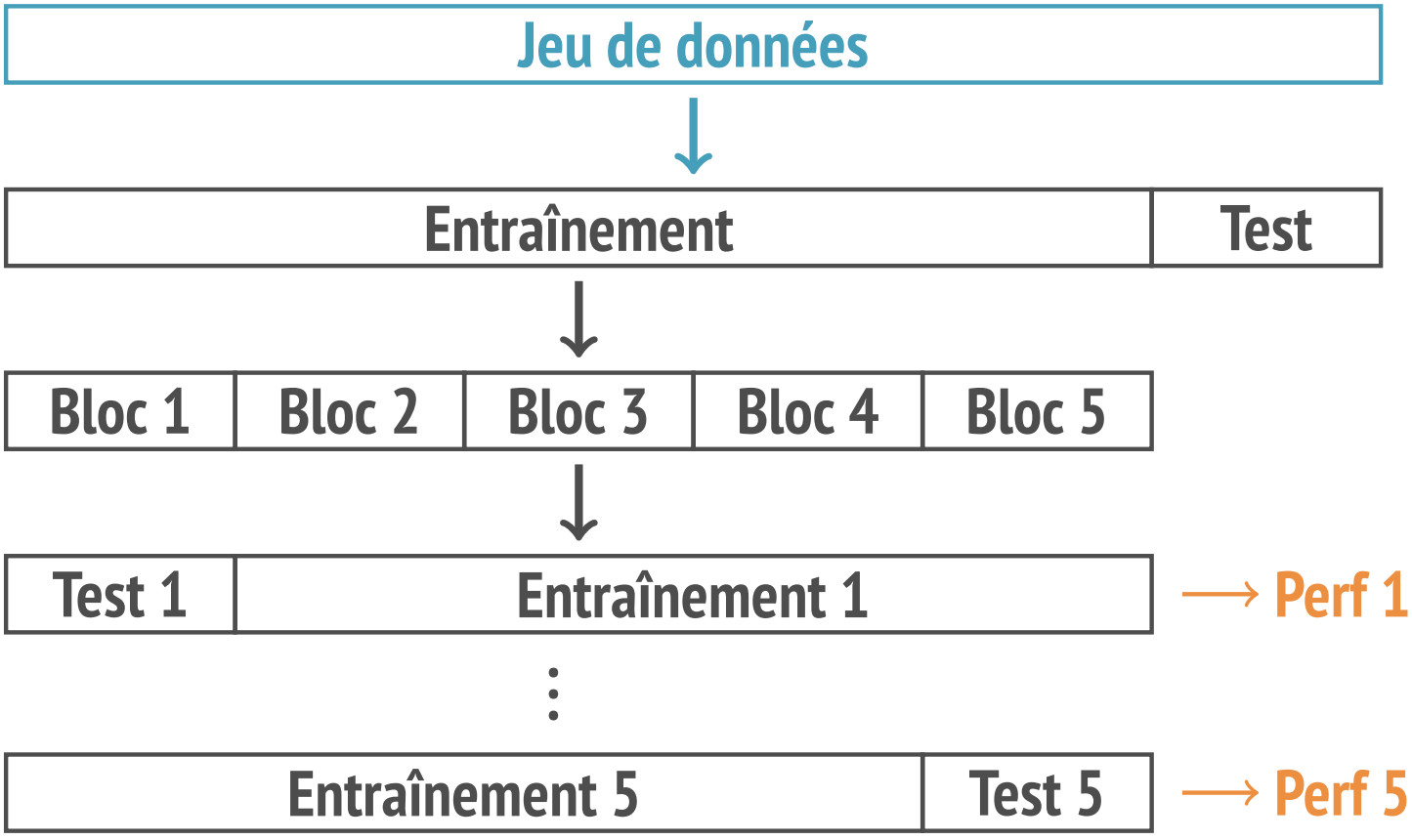
Validation croisée (sélection de modèle)



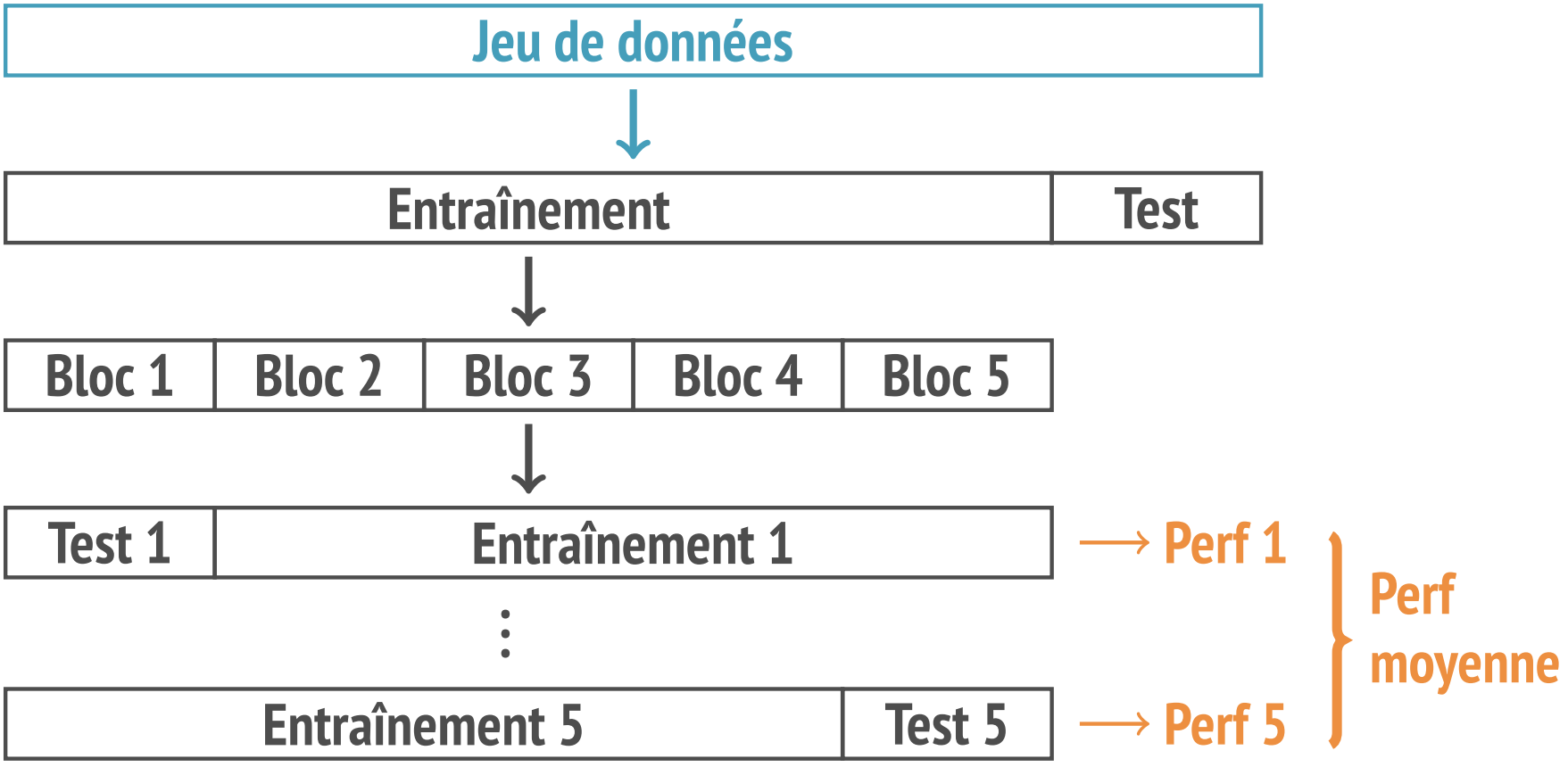
Validation croisée (sélection de modèle)



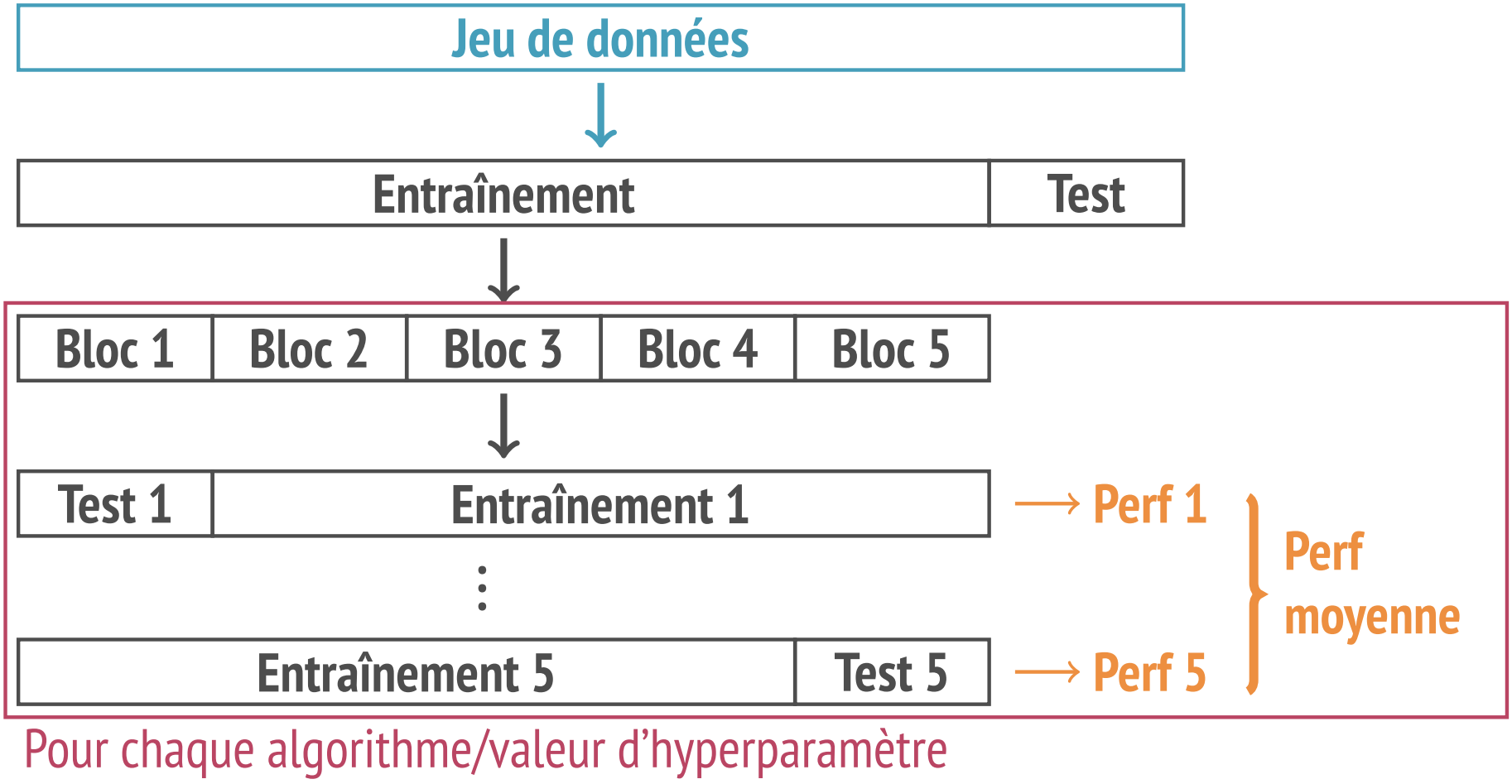
Validation croisée (sélection de modèle)



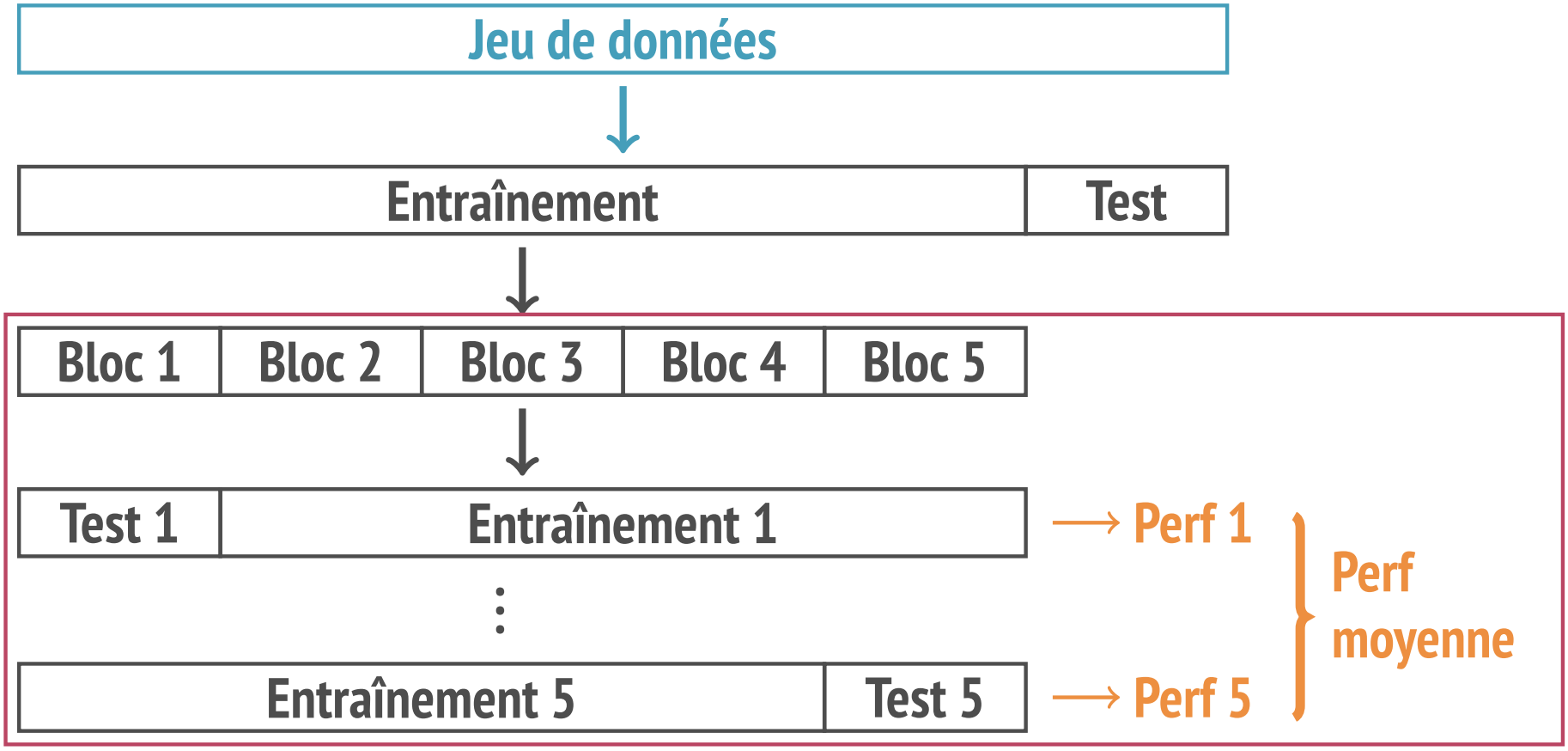
Validation croisée (sélection de modèle)



Validation croisée (sélection de modèle)

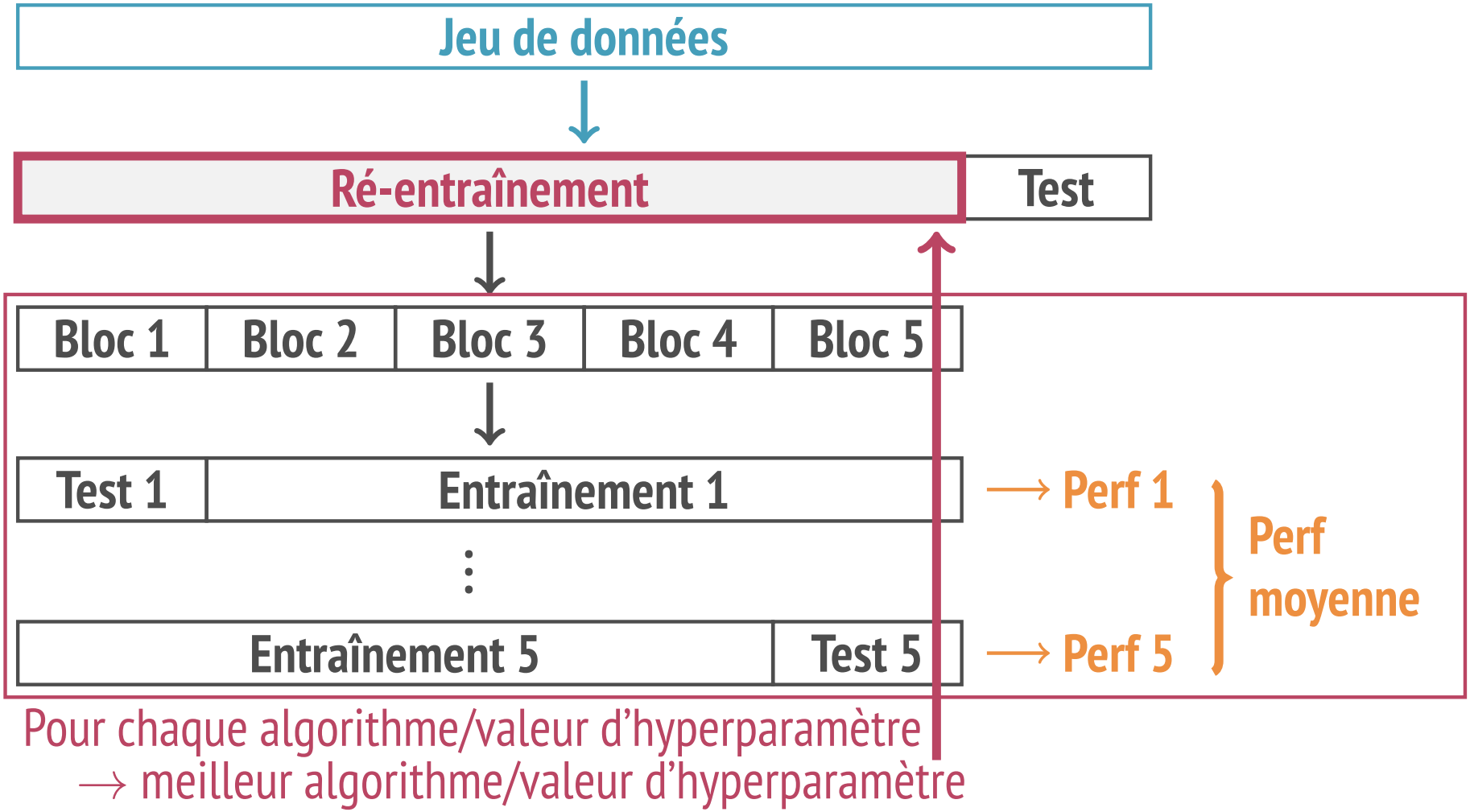


Validation croisée (sélection de modèle)

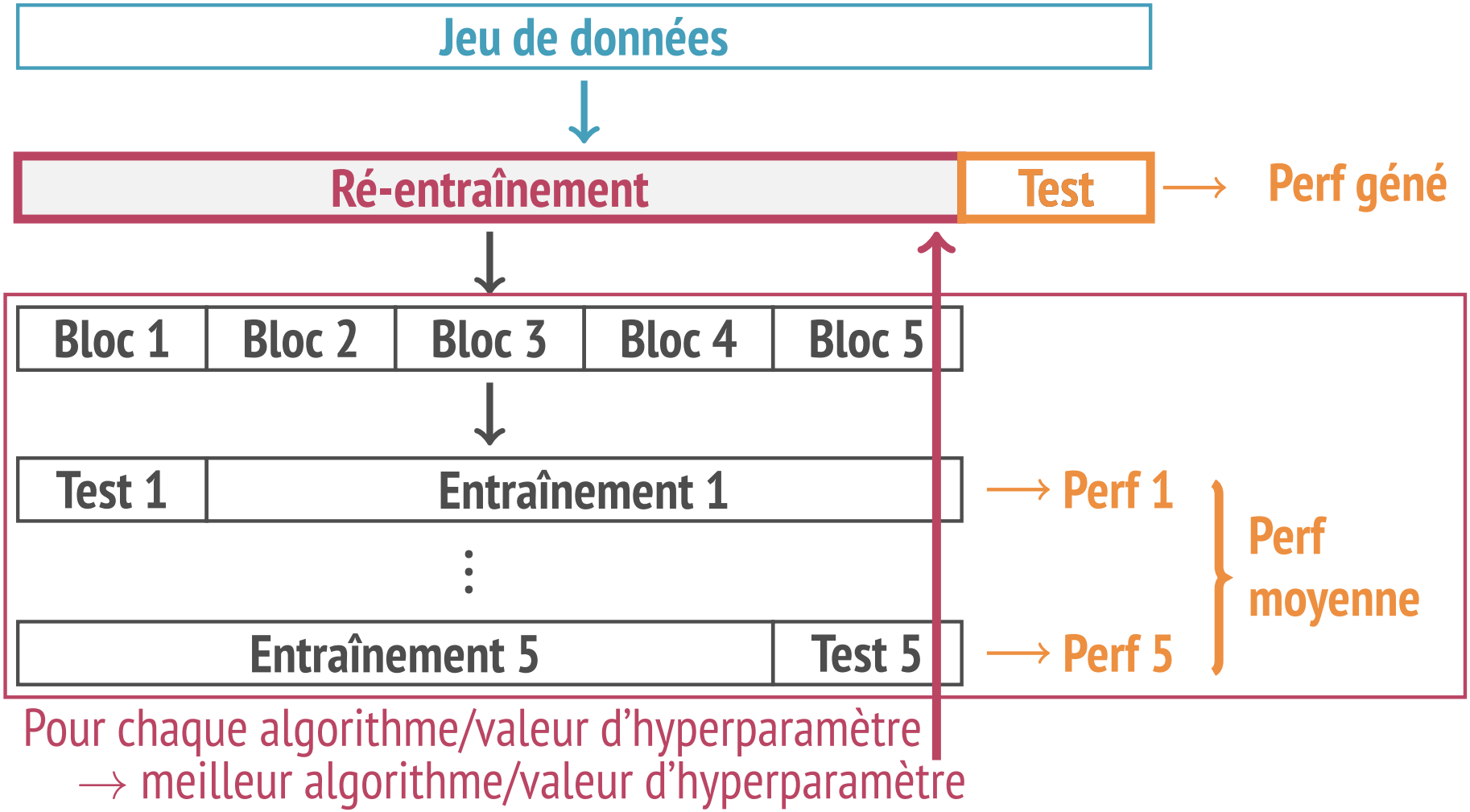


Pour chaque algorithme/valeur d'hyperparamètre
→ meilleur algorithme/valeur d'hyperparamètre

Validation croisée (sélection de modèle)



Validation croisée (sélection de modèle)



Validation croisée imbriquée

- Utiliser une validation croisée :
  - pour **la sélection de modèle** (boucle interne)
  - et pour **l'évaluation** du modèle sélectionné (boucle externe)
- Peut devenir **coûteux en calculs :**
  - **Exemple :** si K = 5 pour la boucle interne, K=10 pour la boucle externe, et on a 10 valeurs d'hyperparamètre à évaluer : 500 modèles à entraîner.
  - Cependant, facile à paralléliser.

2. Régularisation

2.1 Régularisation

- Par principe, la **minimisation du risque empirique** conduit facilement au **surapprentissage**
- La **régularisation** consiste à **contraindre** le problème pour **limiter la complexité** du modèle
  - **Complexité** : notion théorique, par ex. Vapnik–Chervonenkis
  - Intuitivement : « flexibilité » des modèles que l'on peut apprendre
  - Proxys : nombre de paramètres ; amplitude possible de ces paramètres

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(\vec{x}_i)) + \lambda\, \Omega(f) \quad \lambda > 0$$

2.1 Régularisation

- Par principe, la **minimisation du risque empirique** conduit facilement au **surapprentissage**
- La **régularisation** consiste à **contraindre** le problème pour **limiter la complexité** du modèle
  - **Complexité** : notion théorique, par ex. Vapnik–Chervonenkis
  - Intuitivement : « flexibilité » des modèles que l'on peut apprendre
  - Proxys : nombre de paramètres ; amplitude possible de ces paramètres

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(\vec{x}_i)) + \lambda \, \Omega(f) \quad \lambda > 0$$

- **Coefficient de régularisation** $\lambda > 0$: **hyperparamètre** qui gouverne l'importance relative entre la minimisation du risque empirique et la complexité $\Omega(f)$ du modèle.

2.1 Régularisation ridge

- Exemple 1 : **Régularisation ridge** (ou ℓ_2, ou **weight decay**)
  - Pour un modèle paramètrique de paramètres $\theta_1, \theta_2, \ldots, \theta_d$, $\Omega(\vec{\theta}) = \left|\left|\vec{\theta}\right|\right|_2 = \sum_{j=1}^{d} \theta_j^2$ contrôle **l'amplitude des paramètres**

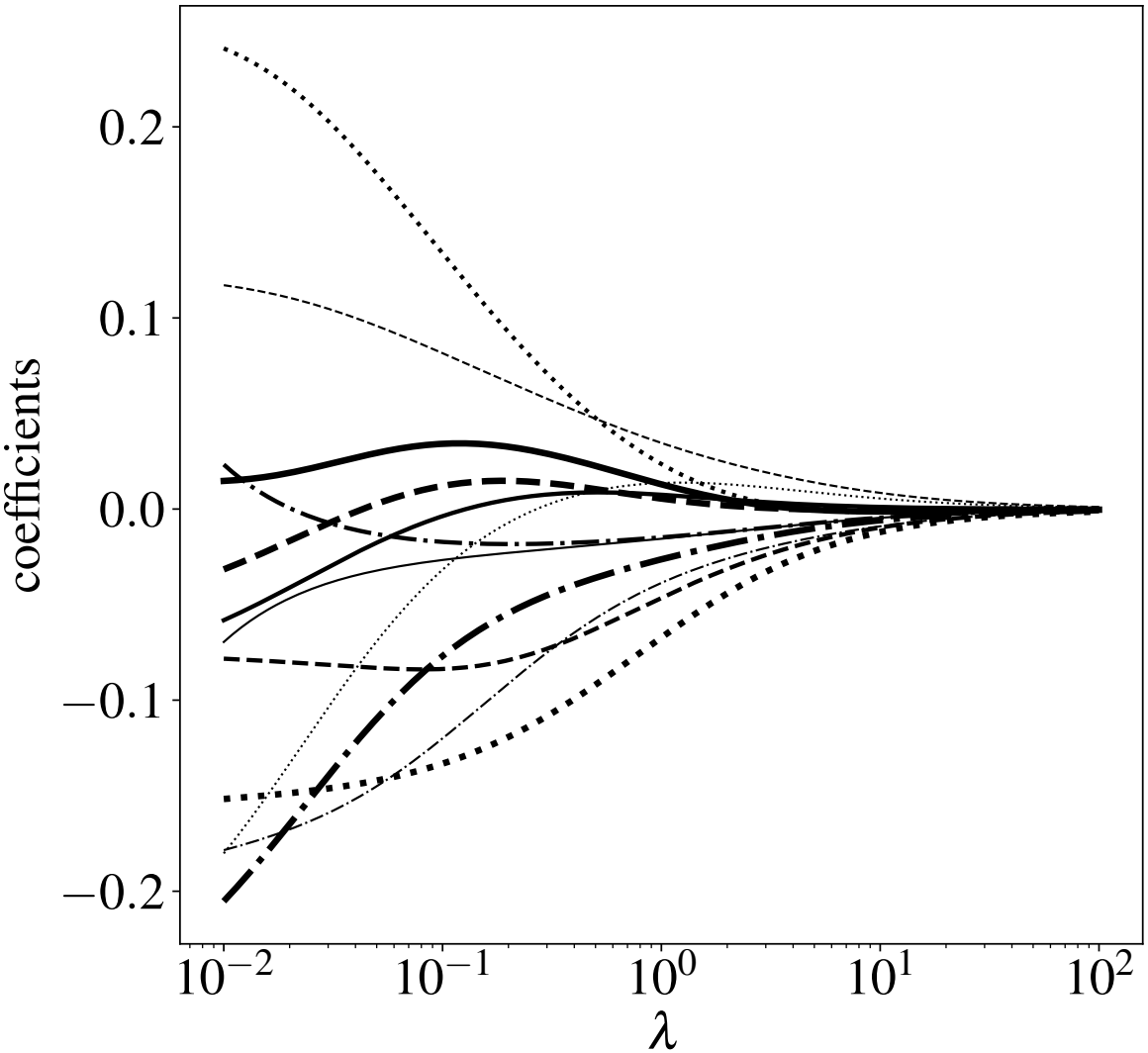
2.1 Régularisation ridge

- Exemple 1 : **Régularisation ridge** (ou ℓ_2, ou **weight decay**)
  - Pour un modèle paramètrique de paramètres $\theta_1, \theta_2, \ldots, \theta_d$, $\Omega(\vec{\theta}) = \left|\left|\vec{\theta}\right|\right|_2 = \sum_{j=1}^{d} \theta_j^2$ contrôle **l'amplitude des paramètres**
  - **Régression ridge :**

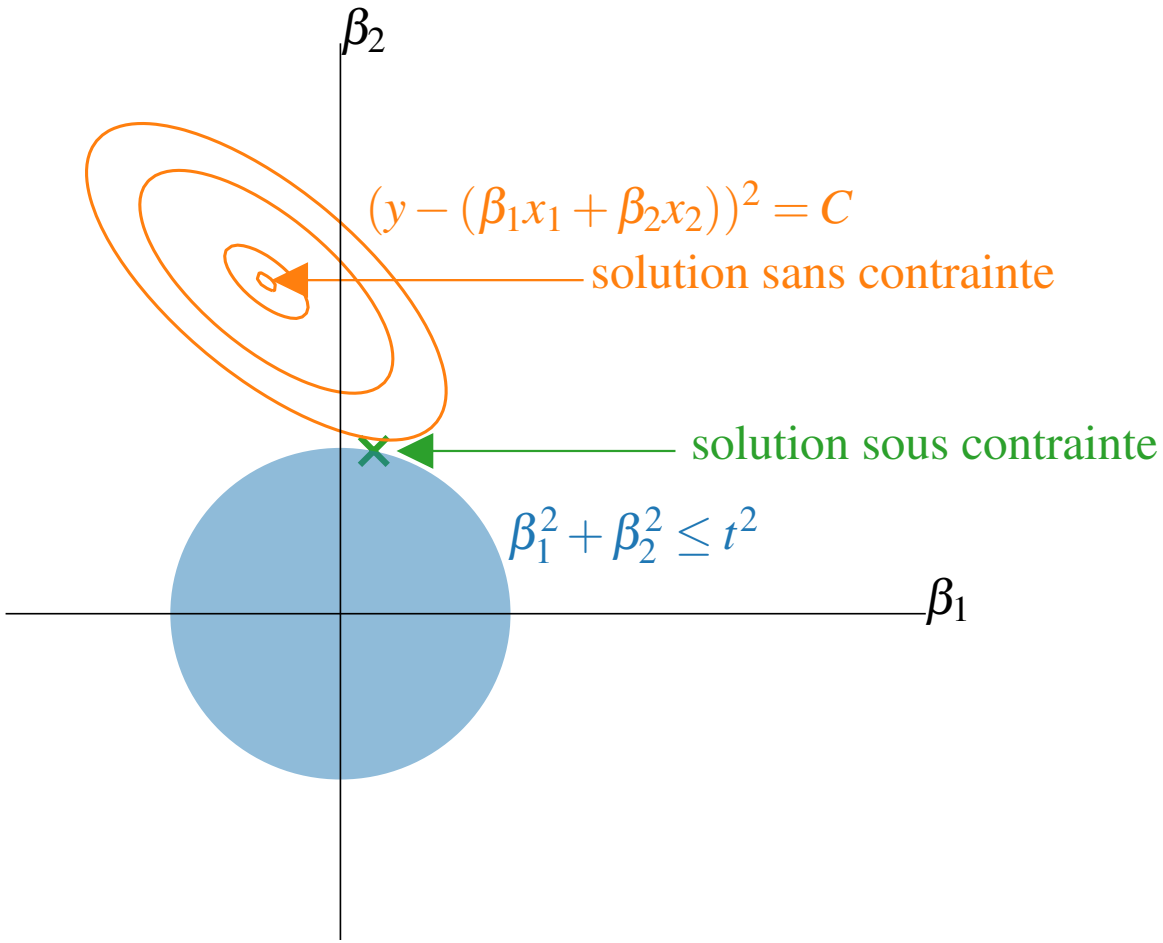
$$\underset{\vec{\beta} \in \mathbb{R}^{p+1}}{\arg\min} \frac{1}{n} \left(\vec{y} - X\vec{\beta}\right)^\top \left(\vec{y} - X\vec{\beta}\right) + \lambda \left|\left|\vec{\beta}\right|\right|_2$$

Admet toujours une unique solution $\vec{\beta}^* = \left(X^\top X + \lambda I_p\right)^{-1} X^\top \vec{y}$ car ajouter une matrice diagonale à coefficients strictement positifs à $X^\top X$ la rend inversible

Chemin de régularisation



Interprétation géométrique



2.2 Lasso

- Exemple 2 : **Régularisation** ℓ_1
  - Pour un modèle paramètrique de paramètres $\theta_1, \theta_2, \ldots, \theta_d$, $\Omega(\vec{\theta}) = \left|\left|\vec{\theta}\right|\right|_1 = \sum_{j=1}^{d} |\theta_j|$ contrôle **le nombre de paramètres non nuls** $\rightarrow$ **parcimonie** (sparsity)

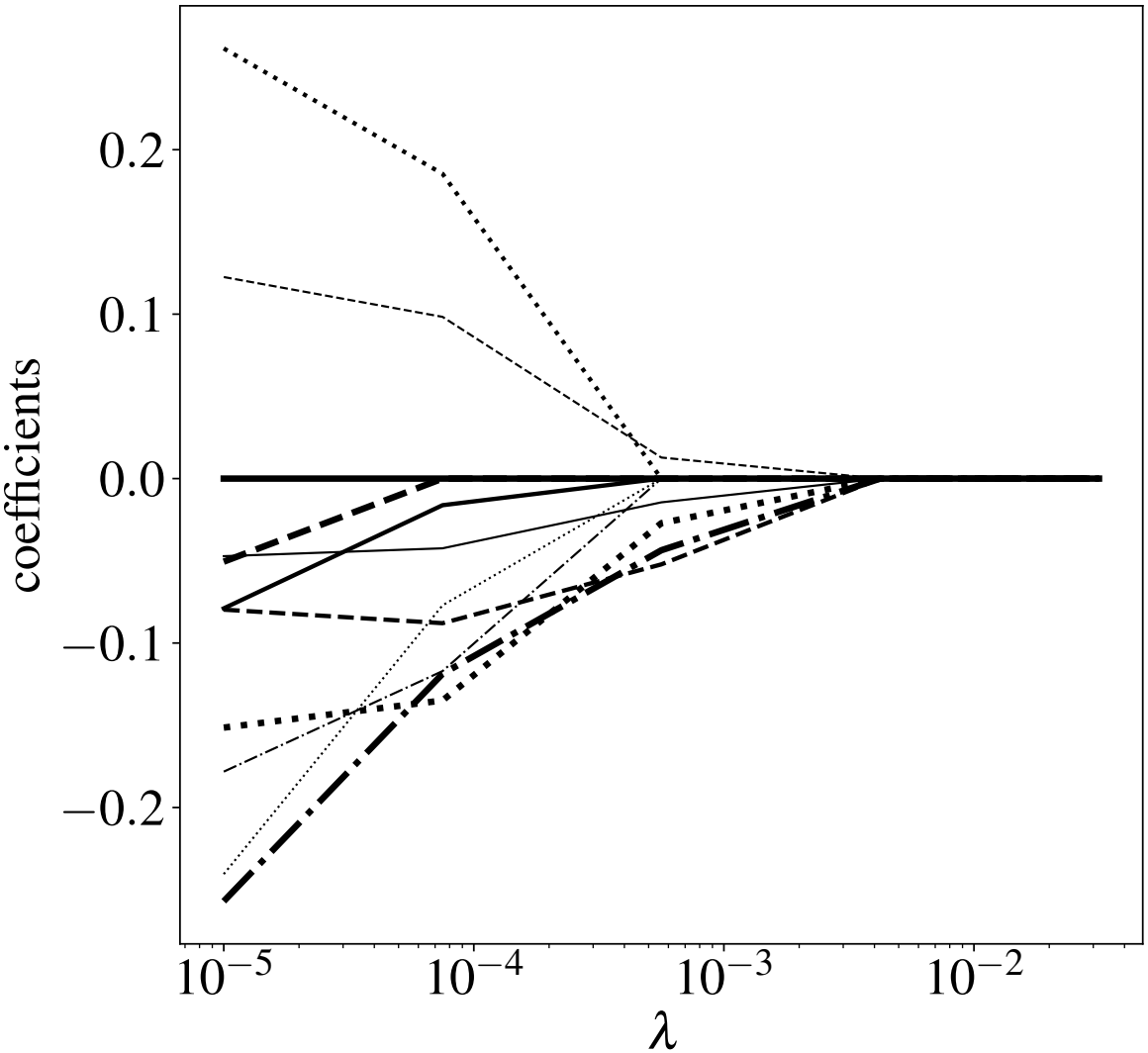
2.2 Lasso

- Exemple 2 : **Régularisation** ℓ_1
  - Pour un modèle paramètrique de paramètres $\theta_1, \theta_2, \ldots, \theta_d$, $\Omega(\vec{\theta}) = \left|\left|\vec{\theta}\right|\right|_1 = \sum_{j=1}^{d} |\theta_j|$ contrôle **le nombre de paramètres non nuls** $\longrightarrow$ **parcimonie** (sparsity)
  - **Lasso** (Least Absolute Sparse Selection Operator) :

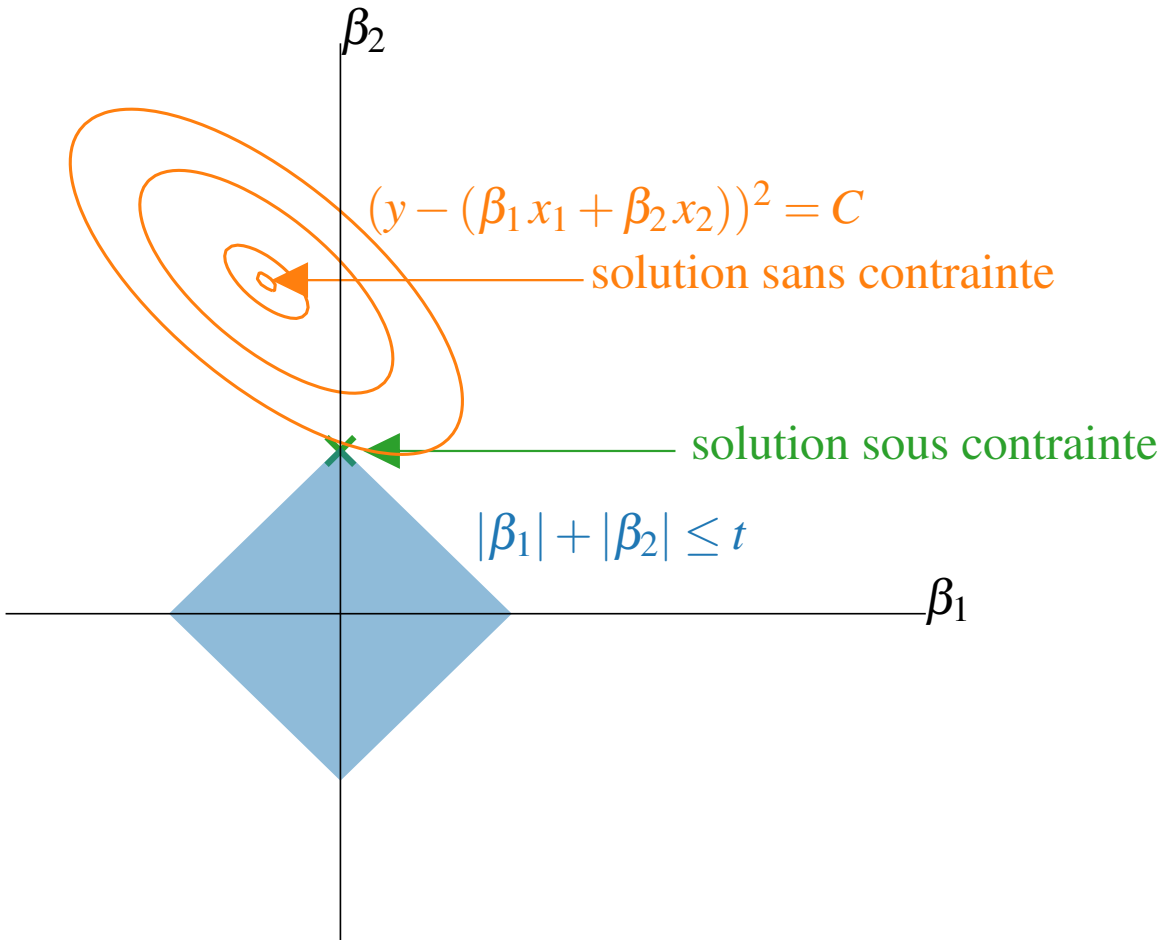
$$\underset{\vec{\beta} \in \mathbb{R}^{p+1}}{\arg\min} \frac{1}{n} \left(\vec{y} - X\vec{\beta}\right)^{\top} \left(\vec{y} - X\vec{\beta}\right) + \lambda \left|\left|\vec{\beta}\right|\right|_1$$

  - Résolu par l'algorithme du gradient.

Chemin de régularisation



Interprétation géométrique



Estimation par maximum a posteriori