

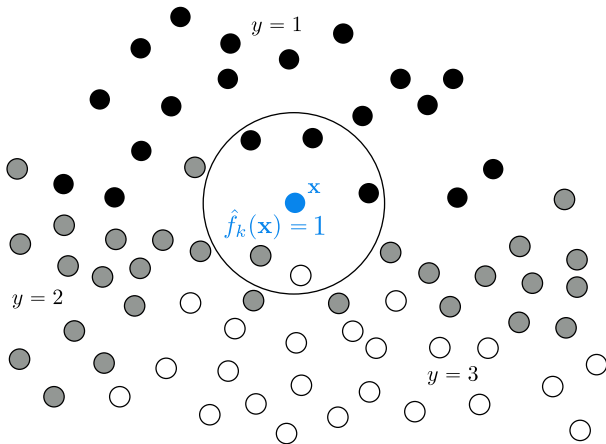
De deux à plusieurs classes

On peut passer du cadre binaire au multi-classe pour toute méthode, e.g., il suffit de tester :

- ▶ “un contre tous” (en : **One-vs.-all**) : créer un classifieur par classe, et produire un score (par exemple une probabilité). Choisir alors la classe avec le score maximum.
- ▶ “un contre un” (en : **One-vs.-one**) : on calcule un classifieur pour toutes les $K(K - 1)/2$ paires. Pour la prédiction on calcule tous les choix possibles et l'on prend la classe qui a reçu le plus de votes.

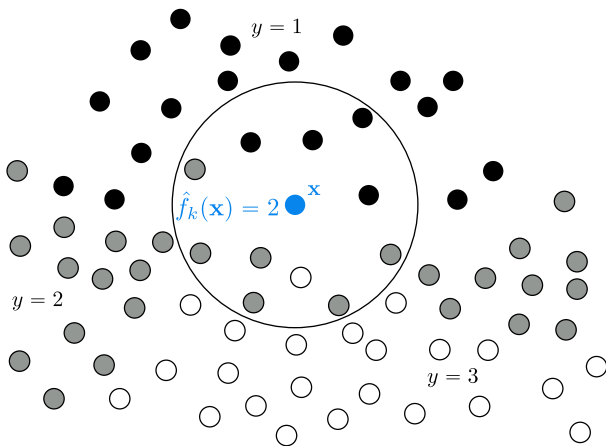
K-nn

Méthode des k -plus proches voisins pour des valeurs du paramètres $k = 5$ et $k = 11$. pour $K = 3$ classes noir ($y = 1$), gris ($y = 2$), blanc ($y = 3$).



K-nn

Méthode des k -plus proches voisins pour des valeurs du paramètres $k = 5$ et $k = 11$. pour $K = 3$ classes noir ($y = 1$), gris ($y = 2$), blanc ($y = 3$).



Avantages / Inconvénients : K-nn

Avantages

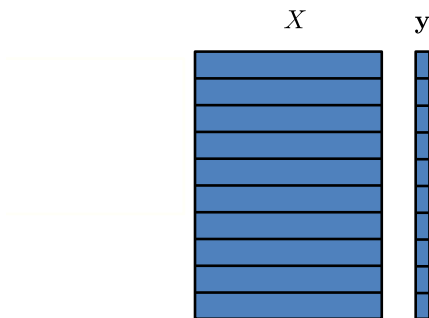
- ▶ séparations non-convexe en général
- ▶ s'adapte avec tout type de distance
- ▶ Multi-classe par défaut.

Inconvénients

- ▶ temps de calcul peut-être long (calculer toute les distances deux à deux, est-ce vraiment utile ?)

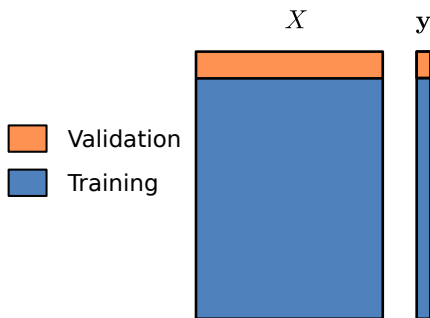
Validation croisée K -fold ($K = 10$)

- ▶ Choisir une grille de taille r de λ à tester : $\lambda_1, \dots, \lambda_r$
- ▶ Diviser (X, Y) selon les observations en K blocs :



Validation croisée K -fold ($K = 10$)

- ▶ Choisir une grille de taille r de λ à tester : $\lambda_1, \dots, \lambda_r$
- ▶ Diviser (X, Y) selon les observations en K blocs :

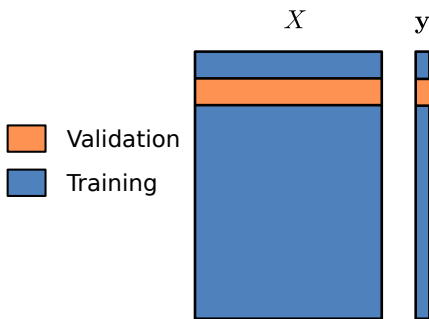


$k = 1$

1. Calculer avec les données d'entraînement les estimateurs pour $\lambda_1, \dots, \lambda_r$: $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
2. Calculer les erreurs de prédiction $\text{Error}_1^k, \dots, \text{Error}_r^k$ sur l'ensemble de validation,

Validation croisée K -fold ($K = 10$)

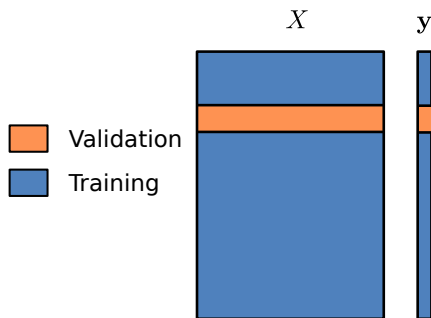
- ▶ Choisir une grille de taille r de λ à tester : $\lambda_1, \dots, \lambda_r$
- ▶ Diviser (X, Y) selon les observations en K blocs :



1. Calculer avec les données d'entraînement les estimateurs pour $\lambda_1, \dots, \lambda_r$: $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
2. Calculer les erreurs de prédiction $\text{Error}_1^k, \dots, \text{Error}_r^k$ sur l'ensemble de validation,

Validation croisée K -fold ($K = 10$)

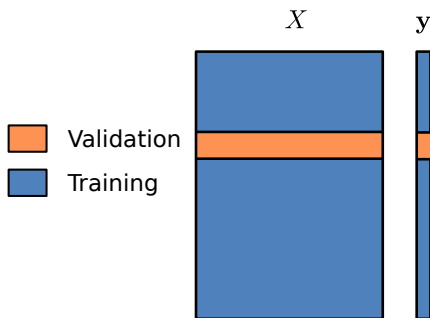
- ▶ Choisir une grille de taille r de λ à tester : $\lambda_1, \dots, \lambda_r$
- ▶ Diviser (X, Y) selon les observations en K blocs :



1. Calculer avec les données d'entraînement les estimateurs pour $\lambda_1, \dots, \lambda_r$: $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
2. Calculer les erreurs de prédiction $\text{Error}_1^k, \dots, \text{Error}_r^k$ sur l'ensemble de validation,

Validation croisée K -fold ($K = 10$)

- ▶ Choisir une grille de taille r de λ à tester : $\lambda_1, \dots, \lambda_r$
- ▶ Diviser (X, Y) selon les observations en K blocs :

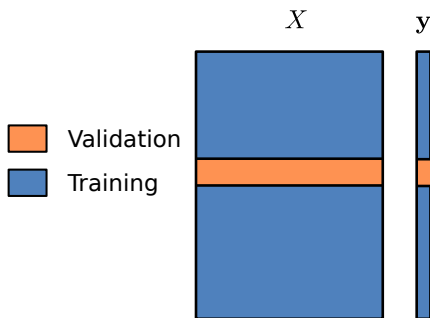


$$k = 4$$

1. Calculer avec les données d'entraînement les estimateurs pour $\lambda_1, \dots, \lambda_r$: $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
2. Calculer les erreurs de prédiction $\text{Error}_1^k, \dots, \text{Error}_r^k$ sur l'ensemble de validation,

Validation croisée K -fold ($K = 10$)

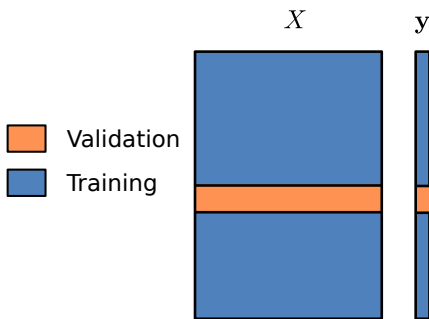
- ▶ Choisir une grille de taille r de λ à tester : $\lambda_1, \dots, \lambda_r$
- ▶ Diviser (X, Y) selon les observations en K blocs :



1. Calculer avec les données d'entraînement les estimateurs pour $\lambda_1, \dots, \lambda_r$: $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
2. Calculer les erreurs de prédiction $\text{Error}_1^k, \dots, \text{Error}_r^k$ sur l'ensemble de validation,

Validation croisée K -fold ($K = 10$)

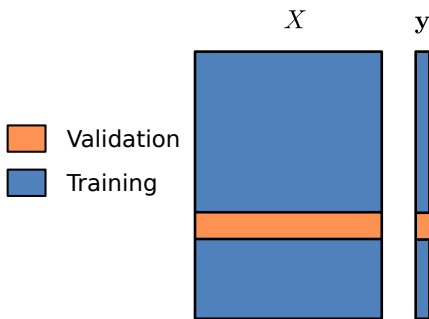
- ▶ Choisir une grille de taille r de λ à tester : $\lambda_1, \dots, \lambda_r$
- ▶ Diviser (X, Y) selon les observations en K blocs :



- $k = 6$
1. Calculer avec les données d'entraînement les estimateurs pour $\lambda_1, \dots, \lambda_r$: $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
 2. Calculer les erreurs de prédiction $\text{Error}_1^k, \dots, \text{Error}_r^k$ sur l'ensemble de validation,

Validation croisée K -fold ($K = 10$)

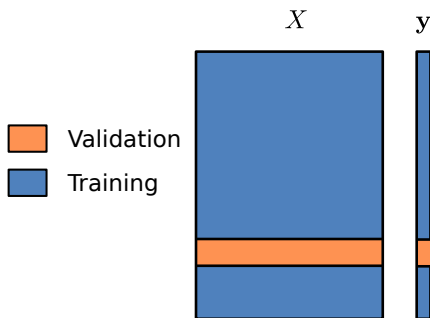
- ▶ Choisir une grille de taille r de λ à tester : $\lambda_1, \dots, \lambda_r$
- ▶ Diviser (X, Y) selon les observations en K blocs :



1. Calculer avec les données d'entraînement les estimateurs pour $\lambda_1, \dots, \lambda_r$: $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
2. Calculer les erreurs de prédiction $\text{Error}_1^k, \dots, \text{Error}_r^k$ sur l'ensemble de validation,

Validation croisée K -fold ($K = 10$)

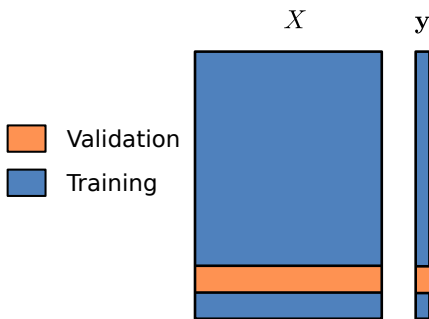
- ▶ Choisir une grille de taille r de λ à tester : $\lambda_1, \dots, \lambda_r$
- ▶ Diviser (X, Y) selon les observations en K blocs :



1. Calculer avec les données d'entraînement les estimateurs pour $\lambda_1, \dots, \lambda_r$: $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
2. Calculer les erreurs de prédiction $\text{Error}_1^k, \dots, \text{Error}_r^k$ sur l'ensemble de validation,

Validation croisée K -fold ($K = 10$)

- ▶ Choisir une grille de taille r de λ à tester : $\lambda_1, \dots, \lambda_r$
- ▶ Diviser (X, Y) selon les observations en K blocs :

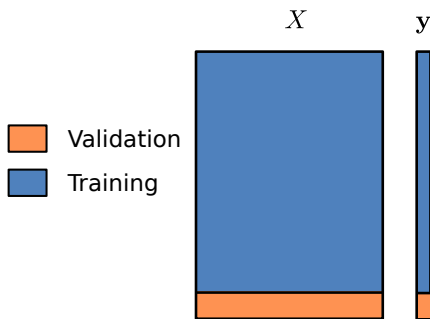


$k = 9$

1. Calculer avec les données d'entraînement les estimateurs pour $\lambda_1, \dots, \lambda_r$: $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
2. Calculer les erreurs de prédiction $\text{Error}_1^k, \dots, \text{Error}_r^k$ sur l'ensemble de validation,

Validation croisée K -fold ($K = 10$)

- ▶ Choisir une grille de taille r de λ à tester : $\lambda_1, \dots, \lambda_r$
- ▶ Diviser (X, Y) selon les observations en K blocs :

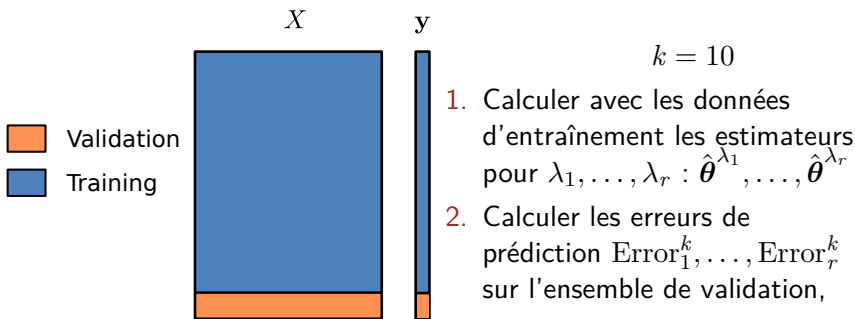


$k = 10$

1. Calculer avec les données d'entraînement les estimateurs pour $\lambda_1, \dots, \lambda_r$: $\hat{\theta}^{\lambda_1}, \dots, \hat{\theta}^{\lambda_r}$
2. Calculer les erreurs de prédiction $\text{Error}_1^k, \dots, \text{Error}_r^k$ sur l'ensemble de validation,

Validation croisée K -fold ($K = 10$)

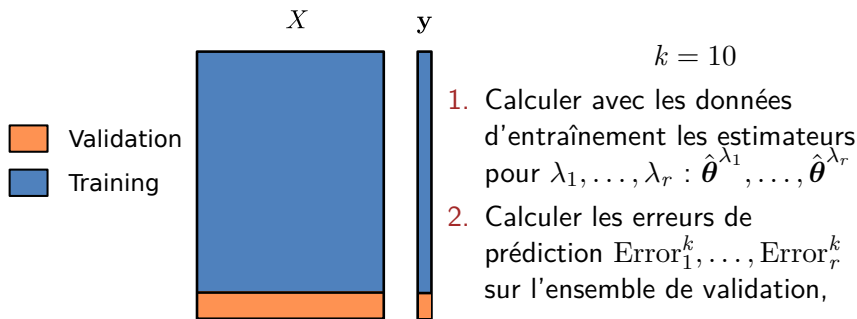
- ▶ Choisir une grille de taille r de λ à tester : $\lambda_1, \dots, \lambda_r$
- ▶ Diviser (X, Y) selon les observations en K blocs :



Choix du paramètre : calculer $\widehat{\text{Error}}_1, \dots, \widehat{\text{Error}}_r$, moyennes des erreurs et choisir $\hat{i}^{\text{CV}} \in \llbracket 1, r \rrbracket$ atteignant la plus petite

Validation croisée K -fold ($K = 10$)

- ▶ Choisir une grille de taille r de λ à tester : $\lambda_1, \dots, \lambda_r$
- ▶ Diviser (X, Y) selon les observations en K blocs :



Choix du paramètre : calculer $\widehat{\text{Error}}_1, \dots, \widehat{\text{Error}}_r$, moyennes des erreurs et choisir $\hat{i}^{\text{CV}} \in \llbracket 1, r \rrbracket$ atteignant la plus petite

Re-calibration : calculer $\hat{\theta}^{\lambda_{\hat{i}^{\text{CV}}}}$ sur l'ensemble des observations

CV en pratique

Cas extrême de validation croisée *cross-validation*

- ▶ $K = 1$ impossible, au moins $K = 2$
- ▶ $K = n$, stratégie “*leave-one-out*” (cf. **Jackknife**) : autant de blocs que de variables

Rem: $K = n$ efficace computationnellement mais instable

Conseils pratiques :

- ▶ “randomiser les observations” : observations dans un ordre aléatoire, évite des blocs de données trop similaires (chaque sous-bloc doit être représentatif de l'ensemble)
- ▶ choix habituels : $K = 5, 10$

Alternatives : partition aléatoire entre ensemble d'apprentissage et validation, version pour séries temporelles, etc.

http://scikit-learn.org/stable/modules/cross_validation.html

Rem: en prédiction on peut aussi moyenner les meilleurs estimateurs obtenus plutôt que de re-calibrer sur toutes les données

Choix de λ : exemple avec $CV = 5$ (I)

