

ML 04 Validation croisée

1 Principe de la validation croisée

Comment choisir la valeur du coefficient de régularisation α pour une régression ridge ? Notons en passant que α est un exemple de ce que l'on appelle un hyperparamètre car sa valeur doit être fixée avant de pouvoir apprendre les paramètres du modèle (dans notre cas, les coefficients d'un modèle linéaire).

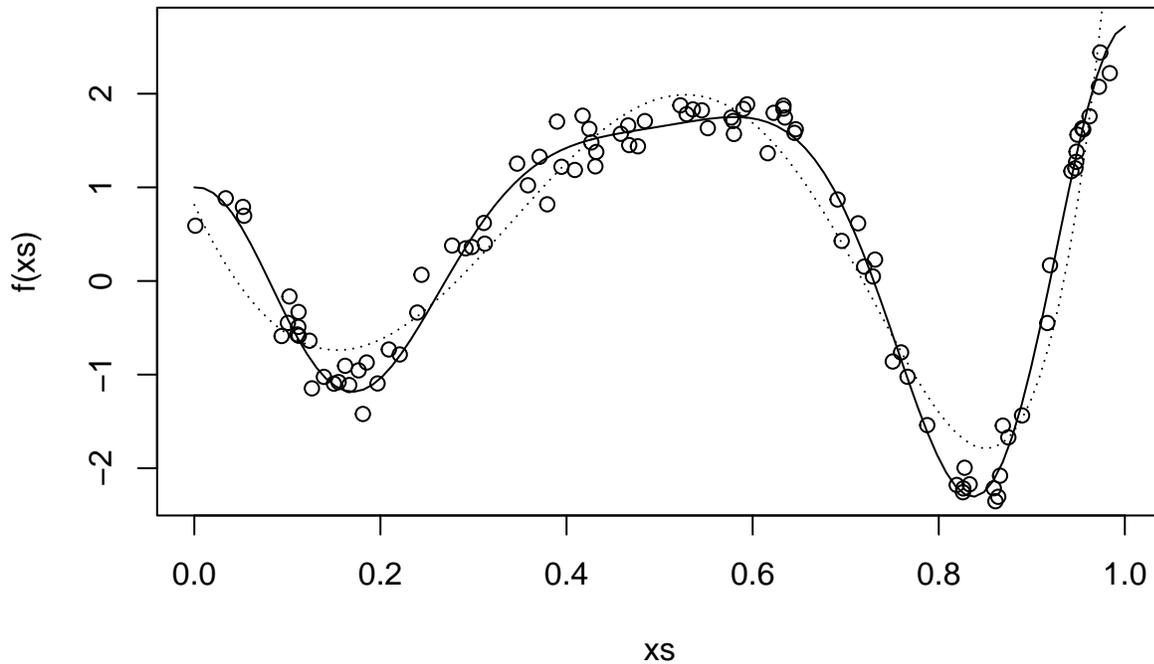
Une possibilité est de diviser le jeu de données en deux parties, l'une utilisée pour apprendre le modèle prédictif, l'autre utilisée pour valider la qualité des prédictions sur des données qui n'ont pas été vues pendant la phase d'apprentissage. On parle de jeu d'entraînement et de jeu de validation ou de test. Cette méthode est appelée "validation croisée".

Il s'agit de tester plusieurs valeurs de l'hyperparamètre α sur le jeu d'entraînement et de conserver celle qui donne les meilleurs résultats sur un jeu de données de validation qui n'a pas été utilisé pour l'entraînement. Nous obtenons ainsi un modèle qui montre souvent une meilleure capacité à prédire sur de nouvelles données.

2 Application de la validation croisée à la régularisation de Tikhonov

Le code source accompagnant ce chapitre comprend une fonction `xvalpart` pour diviser le jeu de données en jeu d'entraînement et jeu de test. Ensuite, la fonction `xvalridge` apprend plusieurs modèles de régression ridge sur le jeu d'entraînement pour des valeurs différentes de l'hyperparamètre α . Elle retourne les coefficients du meilleur modèle avec la moyenne de la valeur absolue des erreurs commises par ce modèle sur le jeu de validation.

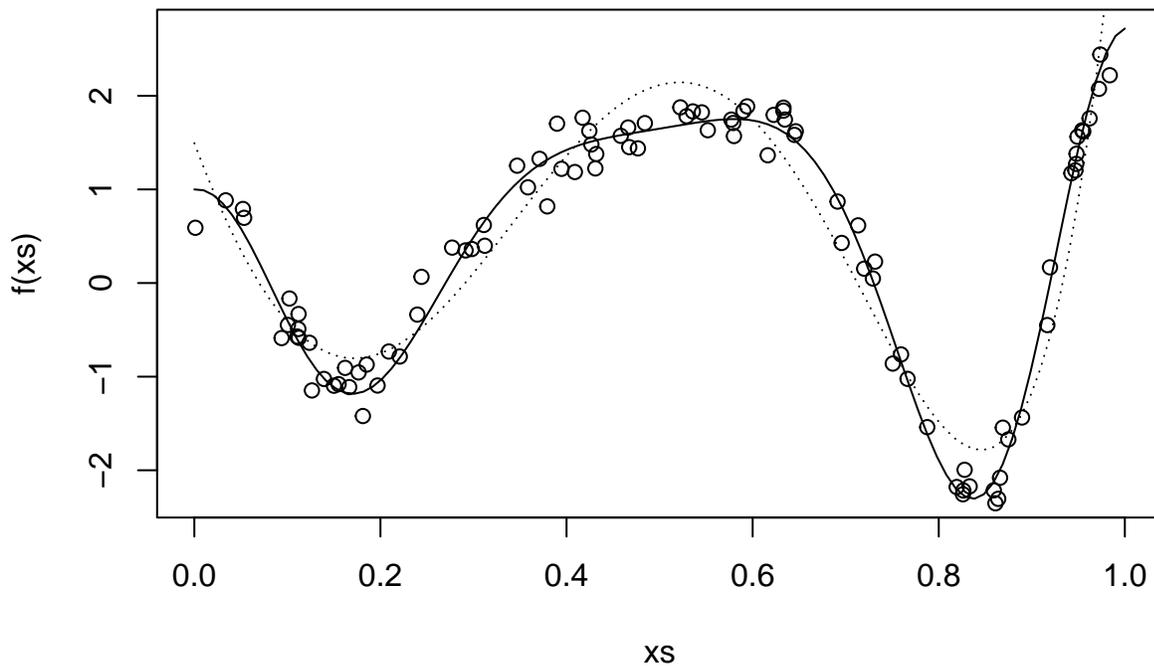
```
set.seed(1123)
N <- 100
deg1 <- 8
data = gendat(N,0.2)
alphas <- c(1E-7, 1E-6, 1E-5, 1E-4, 1E-3, 1E-2, 1E-1, 1)
xvalres <- xvalridge(alphas, data, deg1, 0.8)
plt(data,f)
pltpoly(xvalres$coef)
```



Avec le code ci-dessus, nous générons un nouveau jeu de données composé de 100 observations et nous calculons par validation croisée un polynôme de degré au plus égal à 8 qui modélise au mieux ces données. La valeur de α retenue est : 10^{-4} et la moyenne des erreurs absolues est de 0.3557921.

Testons avec un polynôme de plus petit degré...

```
deg2 <- 5
xvalres <- xvalridge(alphas, data, deg2, 0.8)
plt(data,f)
pltpoly(xvalres$coef)
```



Pour un polynôme de degré 5, la valeur de α retenue est : 10^{-7} et la moyenne des erreurs absolues est de

0.365212. Nous remarquons que pour un espace de fonctions plus riche (polynômes de degré 8 VS polynômes de degré 5), une régularisation plus importante peut-être nécessaire, en particulier pour que le modèle ne soit pas trop sensible au bruit présent dans le jeu d'apprentissage.