Évaluation de modeles

Cette section est basée sur James et al. (2021), chapitre 5.

Nous avons vu dans la section précédente comment mesurer la qualité d'un estimateur : en utilisant l'erreur quadratique moyenne pour une variable quantitative et le taux d'erreur pour une variable qualitative. Cependant, si on calcule l'erreur en utilisant que les données observées, on risque de sous-estimer les vraies erreurs car la fonction $\hat{f}$ a été apprise à l'aide des données observées et donc s'adapte à celles-ci. En effet, on pourrait donc juste prendre un modèle plus flexible qui s'ajustera mieux aux données observées et donc avoir une plus petite valeur de l'erreur. Mais si le modèle s'adapte trop aux données observées, il risque de ne pas être aussi bon si on l'applique sur de nouvelles données. En pratique, on souhaiterait un jeu de données d'**entraînement** sur lequel on apprend le modèle et un jeu de données de **test** sur lequel on calcule l'erreur de prédiction pour ajuster le modèle.

> Remarque : Sur-ajustement et sous-ajustement
>
> Si on choisit à l'aide du jeu de données d'entraînement un modèle trop flexible qui s'ajuste à la partie aléatoire des données de sorte que l'erreur sur un autre jeu de données est plus grande qu'elle ne l'aurait été avec un modèle moins flexible, on parle de **sur-ajustement** (*overfitting*). Dans le cas inverse, si on choisit un modèle rigide, i.e. de sorte que les prédictions soient les mêmes peu importe les données, on parle de **sous-ajustement** (*underfitting*).

Généralement, nous n'avons pas accès à un jeu de données de test pour estimer l'erreur de prédiction sur celui-ci. Dans cette section, on va voir deux méthodes permettant d'estimer l'erreur de prédiction de test en découpant le jeu de données d'entraînement en sous-ensemble et en estimant f sur ces sous-ensemble.

Jeu de données de validation

L'idée du jeu de données de validation est très simple. On divise aléatoirement le jeu de données d'entraînement en deux parties : une partie qui va effectivement servir à entrainer le modèle et l'autre partie qui va servir à estimer l'erreur de prédiction.

Add schema !

Add exemple !

> Comment choisir comment découper le jeu de données ?
>
> La réponse simple : avec de la pratique et une connaissance du domaine. S'il y a beaucoup de données, on peut faire 50-50. Sinon, on peut partir un 70-30.

Désavantage de la méthode :

1. L'erreur de prédiction calculée sur le jeu de données de validation peut être très variable. En effet, elle dépend du nombre d'observations dans le jeu de validation et de quelles sont ses données.
2. On a moins de données pour apprendre le modèle. Comme les modèles ont tendance à moins bien apprendre avec moins de données, l'estimation de l'erreur sur le jeu de validation a tendance à surestimer l'erreur que l'on aurait avec un jeu de test et un modèle appris sur le jeu de données complet.

Validation croisée

Comme l'approche par jeu de données de validation, la validation croisée consiste à faire des sous-ensembles du jeu de données. L'approche consiste à découper de façon aléatoire l'ensemble des observations en K groupes de taille équivalanetes. Le premier sous-ensemble est utilisé comme jeu de données de validation et le modèle est appris sur les $K-1$ autres sous-ensembles. L'erreur de prédiction est calculé sur le premier sous-ensemble. Cette procédure est faite K fois ; à chaque un différent sous-ensemble est utilisé comme jeu de données de validation. À la fin, on a donc K valeurs pour l'erreur de prédiction. On calcule enfin la moyenne des K valeurs de prédiction.

Add schema !

Add exemple !

> Comment choisir le nombre de sous-ensemble ?
>
> En pratique, on utilise $K = 5$ ou $K = 10$. Cela a un avantage computationnel car le modèle doit être appris K fois.

James, Gareth, Daniela Witten, Trevor Hastie, et Robert Tibshirani. 2021. *An Introduction to Statistical Learning : With Applications in R*. Springer Texts in Statistics. New York, NY : Springer US. https://doi.org/10.1007/978-1-0716-1418-1.