STT3030 - Cours #3

Arthur Charpentier

Automne 2024

Types de variables en Apprentissage Supervisé

Les prédicteurs $\mathbf{X}$ et la variable réponse Y peuvent être de différents types:

- qualitative: ordinale (score de préférence) ou nominale (couleur d'une voiture),
- quantitative: discrète (âge en années) ou continue (salaire).

Lorsque Y est qualitative, on parle de classification, et lorsque Y est quantitative, on parle de régression. En général, les algorithmes de *machine learning* disposent à la fois des méthodes de classification et de régression (`RandomForestClassifier` et `RandomForestRegressor` sur scikit-learn en Python).

Exercice 3.7.2: KNN

Carefully explain the differences between the KNN classifier and KNN regression methods.

Dans les deux cas, les k plus proches voisins d'un point de données sont déterminés par un calcul de distance entre les $\mathbf{X}$ (Euclidean, Manhattan, Minkowski). La distinction entre les deux tâches réside dans la manière dont la valeur de Y est attribuée.

1. KNN en classification: vote majoritaire sur les *labels* Y des k plus proches voisins (`KNeighborsClassifier` sur scikit-learn en Python),
2. KNN en régression: moyenne des valeurs de Y des k plus proches voisins (`KNeighborsRegressor` sur scikit-learn en Python).

Exemple: Estimation et évaluation de f (1/3)

On dispose d'une variable réponse Y, le salaire, et une variable explicative X, les heures travaillées. On souhaite réaliser une régression polynomiale.

- **Choix de la forme de** f: Quel modèle apprendre ?

$$Y = \beta_0 + \beta_1 X + \varepsilon \ ?$$
$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \varepsilon \ ?$$
$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_{10} X^{10} + \varepsilon \ ?$$

Pour déterminer l'hyperparamètre d, spécifiant le modèle et donc les paramètres à apprendre, on peut utiliser un échantillon de validation, permettant d'éviter le surapprentissage.

Exemple: Estimation et évaluation de f (2/3)

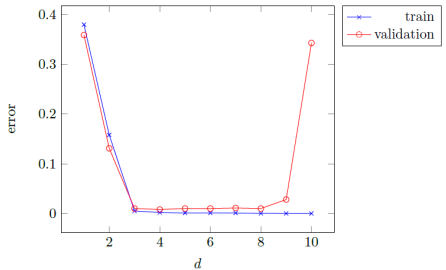


Figure 1: Extrait du livre Shalev-Shwartz and Ben-David (2014)

Si les hyperparamètres étaient choisis uniquement sur l'échantillon d'entraînement, ils pourraient être spécialement adaptés à ses particularités. L'échantillon de validation, distinct, permet d'éviter ce phénomène en évaluant les hyperparamètres sur des données non vues.

Exemple: Estimation et évaluation de f (3/3)

- **Estimation de** f: On doit estimer les paramètres β du modèle $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \beta_4 X^4 + \varepsilon$ en minimisant l'EQM sur les données d'entraînement, $\{(\mathbf{x}_i, y_i) \mid i \in (1, \ldots, n)\}$.
- **Evaluation de** f: Les paramètres estimés $\hat{\beta}$ sur mon échantillon d'entraînement me permettent-ils de bien imiter f pour d'autres échantillons de données ? Pour vérifier la généralisation du modèle estimé sur les données d'entraînement, on calcule l'EQM (ou d'autres métriques) sur un nouvel échantillon, appelé échantillon de test.

Exercice 3.7.4: Régression polynomiale

I collect a set of data ($n = 100$ obs.) containing a single predictor and a quantitative response. I then fit a linear regression model to the data, as well as a separate cubic regression, i.e. $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \varepsilon$.

1. Suppose that the true relationship between X and Y is linear, i.e. $Y = \beta_0 + \beta_1 X + \varepsilon$. Consider the training residual sum of squares (RSS) for the linear regression, and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell ?
2. Answer 1. using test rather than training RSS.
3. Suppose that the true relationship between X and Y is not linear, but we don't know how far it is from linear. Consider the training RSS for the linear regression, and also the training RSS for the cubic regression. Would we expect one to be lower than the other, would we expect them to be the same, or is there not enough information to tell ?
4. Answer 3. using test rather than training RSS.

Modèles linéaires

- Rappel sur la forme du modèle linéaire: on suppose $Y = \mathbf{X}^T\boldsymbol{\beta} + \varepsilon$ où $\boldsymbol{\beta} \in \mathbb{R}^{p+1}$.
- L'hypothèse $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ nous permet de faire de l'inférence sur les paramètres estimés $\hat{\boldsymbol{\beta}} \in \mathbb{R}^{p+1}$.
- Il est possible d'inclure des termes d'interactions $(X_i \times X_j)$ et de la régression polynomiale (X_j^d) pour capturer des effets non linéaires.
- $\mathbf{X}$ peut contenir des variables qualitatives pour lesquelles le logiciel R effectue du *one-hot encoding* et définit une modalité de référence. Les coefficients estimés représentent les effets différentiels par rapport à cette référence.
- Définir une modalité de référence permet d'éviter le problème de colinéarité des variables explicatives qui se produit lorsque des variables sont linéairement dépendantes. Dans ce cas, $\text{rang}(\mathbf{X}) < p + 1$ et donc $\mathbf{X}^T\mathbf{X}$ n'est plus inversible.

Exercice 3.7.3: Modèle linéaire avec intéraction

Suppose we have a data set with five predictors, $X_1 = \text{GPA}$, $X_2 = \text{IQ}$, $X_3 = \text{Gender}$ (1 for Female and 0 for Male), $X_4 = \text{Interaction between GPA and IQ}$, and $X_5 = \text{Interaction between GPA and Gender}$. The response is starting salary after graduation (in thousands of dollars). Suppose we use least squares to fit the model, and get $\beta_0 = 50$, $\beta_1 = 20$, $\beta_2 = 0.07$, $\beta_3 = 35$, $\beta_4 = 0.01$, $\beta_5 = -10$.

1. Which answer is correct, and why?
   - For a fixed value of IQ and GPA, males earn more on average than females.
   - For a fixed value of IQ and GPA, females earn more on average than males.
   - For a fixed value of IQ and GPA, males earn more on average than females provided that the GPA is high enough.
   - For a fixed value of IQ and GPA, females earn more on average than males provided that the GPA is high enough.
2. Predict the salary of a female with IQ of 110 and a GPA of 4.0.
3. True or false: Since the coefficient for the GPA/IQ interaction term is very small, there is very little evidence of an interaction effect. Justify your answer.

Generalized Linear Models (GLM)

- Si le support de Y est différent de $\mathbb{R}$, par exemple $\mathbb{R}^+$, $\mathbb{N}$, $\{0,1\}$; on introduit une fonction de lien g qui permet de se ramener au support souhaité: $g(E[Y]) = \mathbf{X}^T\beta$.
- Pour la régression logistique, avec $\mathcal{Y} = \{0,1\}$, on estime une probabilité $\hat{s}(\mathbf{X}) = \hat{\mathbb{P}}(Y = 1|\mathbf{X}) = \frac{\exp \mathbf{X}^T\hat{\beta}}{1+\exp \mathbf{X}^T\hat{\beta}} \in [0,1]$. Puis la classe prédite $\hat{Y}$ est 1 si $\hat{s}(\mathbf{X}) >$ seuil (par exemple, 0.5), sinon 0.
- Pour la régression multinomiale, avec $\mathcal{Y} = \{1, \cdots, K\}$, on estime K probabilités à l'aide de la fonction *softmax* $\hat{s}_k(\mathbf{X}) = \hat{\mathbb{P}}(Y = k|\mathbf{X}) = \frac{\exp \mathbf{X}^T\hat{\beta}_k}{\sum_{k'=1}^{K} \exp \mathbf{X}^T\hat{\beta}_{k'}} \in [0,1]$. Puis la classe prédite $\hat{Y}$ correspond à celle qui a obtenu le score $\hat{s}_k(\mathbf{X})$ maximal.

La suite: Sélection et Régularisation

- Méthodes de sélection de variables: méthode du meilleur sous-ensemble, méthodes *forward*, *backward* et *stepwise*,
- Méthodes de régularisation: Ridge et Lasso.

Les différentes méthodes vues dans ce cours seront étudiées pour les modèles linéaires mais peuvent également être appliquées à d'autres algorithmes de *machine learning* (parfois sous une appellation différente).

Sélection et Régularisation: Définitions générales

- **Sélection de variables**: processus d'identification et de sélection des prédicteurs les plus pertinents pour un modèle statistique parmi $\mathbf{X}$.
- **Régularisation**: techniques qui contraignent ou pénalisent la complexité d'un modèle (afin d'éviter le surapprentissage).

Sélection de variables: Pourquoi ?

L'objectif de la sélection de modèles est d'éliminer les variables explicatives qui apportent pas ou trop peu d'informations sur la variable à expliquer.

Pourquoi est-il important de faire de la sélection de modèles ?

Quand $n >> p$, il n'y a pas trop de problèmes. Lorsqu'on veut intégrer beaucoup de prédicteurs et donc que p augmente, plusieurs problèmes apparaissent:

- Augmentation de l'incertitude du modèle (plus de paramètres),
- Instabilité des estimateurs (plus grande variance des estimateurs),
- Surapprentissage (si $n = p$ par exemple),
- Modèle plus difficilement interprétable.

Sélection de variables et Modèles linéaires

- Dans le cas des modèles linéaires, lorsque $p > n$, il n'y a plus de solution exacte puisqu'on ne peut plus inverser $\mathbf{X}^T\mathbf{X}$ (le rang de $\mathbf{X}$ ne peut être supérieur à n).
- Si nous avons p prédicteurs, il existe $\frac{p(p-1)}{2!}$ intéractions de premier degré.
- Si nous considérons l'ensemble des polynômes de degré $d = 2$ des prédicteurs, on multiplie p par 2.

Dans le cas des modèles linéaires, en sélectionnant les prédicteurs, on sélectionne le modèle.

Sélection de variables/modèles

On dispose de p prédicteurs $\mathbf{X}$ pour prédire une variable réponse Y. Avant d'entraîner un modèle sur l'ensemble des p prédicteurs, on se demande si l'ensemble des variables est vraiment utile pour la prédiction de Y.

- Pour résoudre ce problème, on procède à la sélection de variables ou plus généralement, la sélection de modèles.
- On souhaite sélectionner le nombre minimal de prédicteurs nous permettant d'obtenir une performance optimale tout en évitant le surapprentissage: compromis entre la complexité du modèle et le nombre d'observations n.

Méthode du meilleur sous-ensemble (1/3)

- Cette méthode consiste à entraîner les modèles contenant tous les 2^p sous-ensembles de variables explicatives possibles, i.e., les modèles obtenus en faisant l'hypothèse que certains coefficients des variables sont égaux à 0,
- Puis on sélectionne le "meilleur modèle" parmi ces 2^p modèles grâce à un critère d'évaluation: EQM, R^2 ajusté, AIC, BIC, etc.

Méthode du meilleur sous-ensemble (2/3)

Algorithm 6.1 *Best subset selection*

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
2. For $k = 1, 2, \ldots p$:
   (a) Fit all $\binom{p}{k}$ models that contain exactly k predictors.
   (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here *best* is defined as having the smallest RSS, or equivalently largest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Figure 2: Algorithme 6.1 du livre de référence James et al. (2013)

Méthode du meilleur sous-ensemble (3/3)

- Cette méthode est très exigeante en terme de temps de calcul (force brute).
- Lorsque p augmente, il faut être en mesure d'entraîner les 2^p modèles.
- Cette méthode n'est pas très efficace: on revoit beaucoup de fois des modèles similaires.
- Il est donc plus efficace d'utiliser une méthode pas à pas afin d'éviter une recherche exhaustive.

Sélection de variables progressive

- La sélection de variables progressive (ou méthodes pas à pas) permet d'évaluer moins de modèles que la méthode exhaustive, donc elle est moins coûteuse en temps de calcul.
- Au cours de ce processus, on ajoute ou on retire une variable à la fois.
- Exemple de James et al. (2013): lorsque $p = 20$, on teste 1,048,576 modèles avec la recherche exhaustive et seulement 211 avec la sélection progressive.

Sélection en avant (*forward*) (1/3)

1. On commence avec un modèle réduit sans variable explicative: $\hat{y} = \bar{y}$, on estime seulement β_0.
2. Puis, on ajoute une variable selon un critère choisi (BIC, AIC, R^2, etc.): on choisit parmi les p modèles possibles $f(\mathbf{x}) = \beta_0 + \beta_j x_j$, $j \in \{1, \cdots, p\}$.
3. Ensuite, on ajoute une autre variable parmi les $p - 1$ variables restantes afin d'obtenir un modèle a 2 variables, et ainsi de suite.
4. L'algorithme s'arrête lorsque le modèle ne s'améliore plus (selon le critère choisi) lorsqu'on ajoute une variable.

Sélection en avant (*forward*) (2/3)

Algorithm 6.2 *Forward stepwise selection*

1. Let $\mathcal{M}_0$ denote the *null* model, which contains no predictors.
2. For $k = 0, \ldots, p-1$:
   (a) Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor.
   (b) Choose the *best* among these $p-k$ models, and call it $\mathcal{M}_{k+1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Figure 3: Algorithme 6.2 du livre de référence James et al. (2013)

Sélection en avant (*forward*) (3/3)

Un inconvénient de cette méthode est que, lorsqu'une variable est ajoutée au modèle, elle ne peut plus être retirée, ce qui peut parfois conduire à des modèles incluant des variables non significatives (en cas de corrélation avec une autre variable par exemple).

Sélection en arrière (*backward*) (1/2)

1. On commence par le modèle linéaire complet contenant toutes les p variables explicatives.
2. Ensuite, à chaque étape, on retire la variable explicative qui a le moins d'impact sur la performance du modèle, selon le critère de sélection choisi.
3. Après chaque suppression, on réévalue le modèle avec $p-1$ variables explicatives.
4. Ce processus est répété jusqu'à ce que la suppression d'une variable entraîne une dégradation de la performance du modèle, ou jusqu'à ce qu'un critère d'arrêt prédéfini soit atteint.

Sélection en arrière (*backward*) (2/2)

Algorithm 6.3 *Backward stepwise selection*

1. Let $\mathcal{M}_p$ denote the *full* model, which contains all p predictors.
2. For $k = p, p-1, \ldots, 1$:
   (a) Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   (b) Choose the *best* among these k models, and call it $\mathcal{M}_{k-1}$. Here *best* is defined as having smallest RSS or highest R^2.
3. Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2.

Figure 4: Algorithme 6.3 du livre de référence James et al. (2013)

Méthode mixte (*stepwise*)

Cette méthode combine les méthodes de sélection pas à pas *forward* **et** *backward*.

Après chaque itération de la méthode de sélection en avant, on effectue une étape d'élimination en arrière: on vérifie donc que l'ajout d'une variable ne rende pas non significative une autre. Et, on répète les deux opérations successivement.

On peut également commencer par l'algorithme *backward* puis utiliser l'algorithme *forward*.

Méthodes progressives: Conclusion

- Une vieille stratégie.
- Très peu utilisé en pratique maintenant (sauf à petit échelle, $+/-$ une variable).
- Maintenant, ces méthodes sont remplacées par des approches de régularisation.

Régularisation: Cas général (1/3)

Compromis biais-variance: $\text{MSE} = \textit{Bias}(\hat{f}) + \textit{Var}(\hat{f}) + \sigma^2$

- **Variance** de $\hat{f}$: est-ce que ma prédiction $\hat{f}$ varie beaucoup lorsqu'on change le *train set* ?
- **Biais** de $\hat{f}$: différence entre la prédiction moyenne et la valeur cible, pouvant être due à des hypothèses simplificatrices dans le modèle.
- **Erreur irréductible**: erreur ne pouvant être réduite par un modèle (facteurs non mesurés, bruit dans les données).

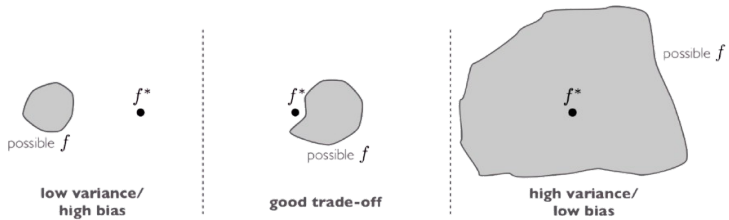


Figure 5: Extrait du cours Courville (2024)

Documentation: `https://allenkunle.me/bias-variance-decomposition`

Régularisation: Cas général (2/3)

Les méthodes de régularisation permettent de trouver un compromis biais-variance afin d'améliorer la capacité de généralisation du modèle. Ainsi :

- En imposant des contraintes sur le modèle, on augmente le biais du modèle tout en réduisant la variance.
- Une méthode efficace de régularisation est celle qui réussit à obtenir un compromis favorable, en diminuant significativement la variance sans trop augmenter le biais.

Régularisation: Cas général (3/3)

La stratégie moderne de sélection de modèles et/ou de variables repose principalement sur l'ajout d'un terme supplémentaire dans la fonction objective, c'est-à-dire la fonction à optimiser lors de l'apprentissage statistique.

Soit $L(\boldsymbol{\theta}; \mathbf{X}, Y)$ la fonction objective et $\boldsymbol{\theta}$ les paramètres du modèle qu'on souhaite apprendre.

La régularisation consiste à introduire un terme additionnel dans la fonction objective pour contrer ou pénaliser certains aspects du modèle. On obtient donc une nouvelle fonction de perte:

$$\begin{aligned}\tilde{L}(\boldsymbol{\theta}; \mathbf{X}, Y) &= L(\boldsymbol{\theta}; \mathbf{X}, Y) + \text{Pénalité} \\ &= L(\boldsymbol{\theta}; \mathbf{X}, Y) + \lambda\Omega(\boldsymbol{\theta}) \quad .\end{aligned}$$

avec $\lambda \in \mathbb{R}$ un hyperparamètre qui équilibre l'effet du terme de régularisation et $\Omega(\boldsymbol{\theta})$ le régularisateur utilisé.

Régularisation, pénalisation et contraction: Modèle linéaire (1/6)

Dans le cas d'un modèle linéaire avec p prédicteurs $\mathbf{X}$ et la variable réponse Y, on cherche un compromis entre ces deux modèles extrêmes:

- Le modèle simple $\hat{y} = \bar{y}$ pour lequel on estime seulement β_0 et $\beta_1 = \beta_2 = \cdots = \beta_p = 0$.
- Le modèle complet $\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$.

Régularisation, pénalisation et contraction: Modèle linéaire (2/6)

Si l'on dispose de n observations et p prédicteurs, la variance des estimateurs des moindres carrés et des prévisions est liée à la matrice $(\mathbf{X}^T\mathbf{X})^{-1}$.

Or, la matrice $\mathbf{X}^T\mathbf{X}$ n'est pas inversible lorsque $p > n$ ou lorsque les covariables ne sont pas linéairement indépendantes.

En pratique, lorsque les variables explicatives sont fortement corrélées entre elles (problème de colinéarité), ou lorsque le nombre de variables explicatives p est du même ordre de grandeur que n, alors la matrice $\mathbf{X}^T\mathbf{X}$ est mal conditionnée et certains coefficients de la matrice $(\mathbf{X}^T\mathbf{X})^{-1}$ deviennent grand, ce qui conduit à des estimateurs avec une forte variance.

Régularisation, pénalisation et contraction: Modèle linéaire (3/6)

Si l'on veut réduire les problèmes liés à une grande valeur de p, on veut pousser le modèle complet à ressembler au modèle simple.

D'une certaine manière, on veut donc pousser les β_j, $j \in \{1, \cdots, p\}$, vers 0.

On veut donc faire de la contraction sur les coefficients β_j, $j \in \{1, \cdots, p\}$, afin de pénaliser de trop grands coefficients (on réduit donc leur variance).

Régularisation, pénalisation et contraction: Modèle linéaire (4/6)

Rappelons-nous le problème d'optimisation pour l'entraînement du modèle complet. On cherche les coefficients $(\beta_0, \boldsymbol{\beta}) \in \mathbb{R} \times \mathbb{R}^p$ minimisant l'erreur de prédiction, correspondant à la fonction objective:

$$L(\beta_0, \boldsymbol{\beta}) = \text{RSS} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{i,j})^2$$

En résolvant ce problème d'optimisation, on obtient la droite des moindres carrés:

$$(\hat{\beta}_0, \hat{\boldsymbol{\beta}}) = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

Ce résultat nous donne généralement des coefficients $\hat{\boldsymbol{\beta}}$ bien différents de 0.

Régularisation, pénalisation et contraction: Modèle linéaire (5/6)

La régularisation consiste à ajouter une pénalité dans la fonction objective afin de contrer ou de pénaliser certains aspects du modèle, i.e. allant à l'encontre des autres termes à optimiser.

$$\begin{aligned}\tilde{L}(\beta_0, \boldsymbol{\beta}) &= \text{RSS} + \text{Pénalité} \\ &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{i,j})^2 + \lambda\Omega(\boldsymbol{\beta}) \quad .\end{aligned}$$

Régularisation, pénalisation et contraction: Modèle linéaire (6/6)

Les pénalités présentées par la suite ne s'appliquent pas à β_0 mais seulement à $(\beta_1, \beta_2, \cdots, \beta_p)$.

En effet, on souhaite réduire l'impact estimé de chaque variable sur la réponse, ce qui ne concerne pas β_0 représentant la prédiction de la réponse lorsque $x_{i,1} = x_{i,2} = \cdots = x_{i,p} = 0$: $\hat{\beta}_0 = \frac{1}{n}\sum_{i=1}^{n} y_i - \sum_{j=1}^{p}\left(\frac{1}{n}\sum_{i=1}^{n}\hat{\beta}_j x_{i,j}\right)$.

Dans le cas où les variables $\mathbf{X}$ ont été centrées avant d'appliquer la régression, $\hat{\beta}_0 = \frac{1}{n}\sum_{i=1}^{n} y_i$, que ce soit avec ou sans pénalité.

Régression Ridge (1/6)

Pour réduire la variance des estimateurs, la régression Ridge introduit un terme de pénalité pour pénaliser les fortes valeurs de $\hat{\boldsymbol{\beta}}$:

$$\Omega(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_2 = \sum_{j=1}^{p} \beta_j^2 \quad .$$

Ainsi, l'estimateur Ridge se définit par:

$$\hat{\boldsymbol{\beta}}_{\text{Ridge}} = \underset{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}\times\mathbb{R}^p}{\text{argmin}} \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{i,j})^2 + \lambda\sum_{j=1}^{p}\beta_j^2$$

Pour le modèle complet, RSS est minimisé avec des coefficients $\boldsymbol{\beta}$ généralement différents de 0, alors que le terme de pénalité est minimisé lorsque tout les coefficients $\boldsymbol{\beta}$ sont tous égaux à 0.

Régression Ridge (2/6)

Plus généralement, en *machine learning*, la pénalité de la régression Ridge s'utilise sous le nom de *L2 regularization*, *weight decay*.

Régression Ridge (3/6)

Le paramètre λ est un hyperparamètre, donc un paramètre que nous devons déterminer au prélable, à l'aide d'un échantillon de validation par exemple.

- Plus λ est grand, plus on pénalise de grands coefficients $\hat{\boldsymbol{\beta}}$ donc plus on aura une solution simple, près de $\hat{y} = \hat{\beta}_0 = \bar{y}$, (avec de petits coefficients $\hat{\boldsymbol{\beta}}$).
- Plus λ est petit, plus le modèle obtenu est proche du modèle linéaire standard. Dans le cas où $\lambda = 0$, alors on retrouve le modèle complet.

Ainsi, λ apparaît comme notre paramètre de réglage du compromis entre biais et variance.

Regression Ridge (4/6)

- Il existe une solution exacte au problème de régression Ridge, i.e., on peut trouver le minimum de manière calculatoire pour chaque valeur de λ. La somme des carrés des résidus est:

$$S(\beta) = (\mathbf{Y} - \mathbf{X}\beta)^T(\mathbf{Y} - \mathbf{X}\beta) + \lambda\beta^T\beta$$

$$\begin{aligned}\mathbf{0} = \frac{\partial S}{\partial \beta}(\hat{\beta}) &= \frac{\partial}{\partial \beta}(\mathbf{Y}^T\mathbf{Y} - \beta^T\mathbf{X}^T\mathbf{Y} - \mathbf{Y}^T\mathbf{X}\beta + \beta^T\mathbf{X}^T\mathbf{X}\beta + \beta^T\beta)\Big|_{\beta=\hat{\beta}} \\ &= -2\mathbf{X}^T\mathbf{Y} + 2\mathbf{X}^T\mathbf{X}\hat{\beta} + 2\lambda\mathbf{I}\hat{\beta}\end{aligned}$$

Ainsi, $\hat{\beta}_{\text{Ridge}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}$ (on s'éloigne des valeurs propres de $\mathbf{X}^T\mathbf{X}$ égales à 0 en ajoutant le terme $\lambda\mathbf{I}$).

Régression Ridge (5/6)

- Méthode *shrinkage*: le modèle contracte tous les paramètres en simultané.

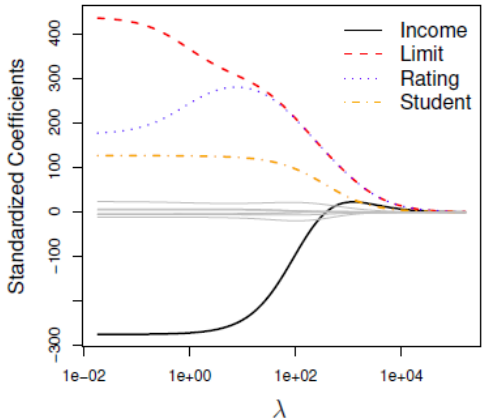


Figure 6: Extrait de la figure 6.4 du livre James et al. (2013)

Regression Ridge (6/6)

- λ permet d'obtenir un meilleur compromis entre biais et variance. En réduisant la variance, λ permet de faire évoluer le modèle de manière progressive sur le spectre biais-variance.

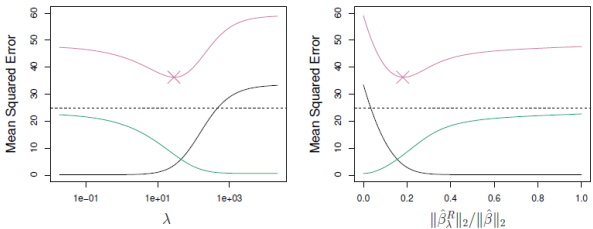


FIGURE 6.5. *Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set, as a function of* λ *and* $\|\hat{\beta}^R_\lambda\|_2/\|\hat{\beta}\|_2$. *The horizontal dashed lines indicate the minimum possible MSE. The purple crosses indicate the ridge regression models for which the MSE is smallest.*

Figure 7: Figure 6.5 du livre James et al. (2013)

Régression Lasso (1/3)

La régression Ridge est utile pour réduire la variance des estimateurs et des prévisions lorsque la matrice $\mathbf{X}^T\mathbf{X}$ est mal conditionnée. Cependant, toutes les variables sont conservées dans le modèle. La régression Lasso résout ce problème en utilisant la norme L1 et opère donc de la sélection de variables en rendant des coefficients β_j, $j \in \{1, \cdots, p\}$, égaux à 0.

La régression Lasso introduit le terme de pénalité suivant:

$$\Omega(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |\beta_j| \quad .$$

Ainsi, l'estimateur Lasso se définit par:

$$\hat{\boldsymbol{\beta}}_{\text{Lasso}} = \underset{(\beta_0,\boldsymbol{\beta})\in\mathbb{R}\times\mathbb{R}^p}{\text{argmin}} \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{i,j})^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

Régression Lasso (2/3)

- Plus généralement, en *machine learning*, la pénalité de la régression Lasso s'utilise sous le nom de *L1 regularization*.
- Le paramètre λ est une nouvelle fois un hyperparamètre à déterminer au prélable. Plus λ est grand, plus l'importance de la pénalité est grande.
- Il n'existe pas de solution explicite au problème de régression Lasso. En raison de la non-différentiabilité de la fonction objective due à la norme L1, des algorithmes numériques de type "coordinate descent" sont utilisés pour résoudre le problème d'optimisation (Friedman et al. (2010)).

Régression Lasso (3/3)

- Le modèle pousse réellement certains coefficients à 0 grâce à la norme L1.

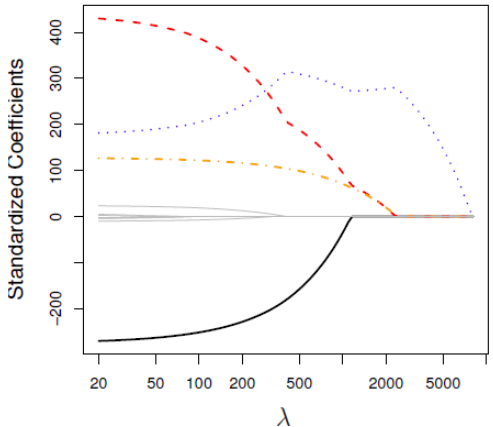


Figure 8: Extrait de la figure 6.6 du livre James et al. (2013)

Ridge et Lasso: Optimisation sous contrainte (1/3)

Les estimateurs Ridge et Lasso peuvent être vus comme des solutions à des problèmes duals (méthode lagrangienne), i.e., on peut les reformuler en tant que problèmes d'optimisation sous contrainte.

$$\hat{\boldsymbol{\beta}}_{\text{Ridge}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\text{argmin}} \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{i,j})^2 \text{ s.t. } \sum_{j=1}^{p}\beta_j^2 \leq s$$

$$\hat{\boldsymbol{\beta}}_{\text{Lasso}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\text{argmin}} \frac{1}{n}\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{i,j})^2 \text{ s.t. } \sum_{j=1}^{p}|\beta_j| \leq s$$

$\rightarrow$ A chaque valeur de pénalité λ, correspond un certain $s \geq 0$.

Ridge et Lasso: Optimisation sous contrainte (2/3)

Maintenant, voyons voir comment la forme de la contrainte affecte les solutions du problème de minimisation.

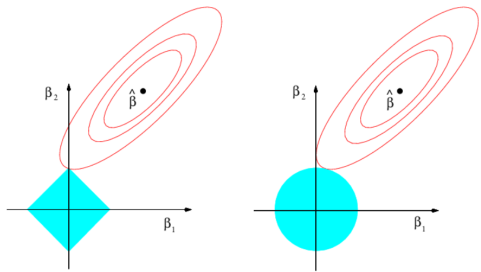


FIGURE 6.7. *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$, *while the red ellipses are the contours of the RSS.*

Figure 9: Figure 6.7 du livre James et al. (2013)

Ridge et Lasso: Optimisation sous contrainte (3/3)

- La forme de l'espace bornée par la contrainte affecte le type de solutions que nous obtenons.
- L'intersection entre les minimums pour le RSS et l'espace de contrainte peut être sur les coins créés par la pénalité pour la norme L_1.
- Notez que ces coins apparaitront sur toutes normes L_p pour $p \in [1, 2)$.

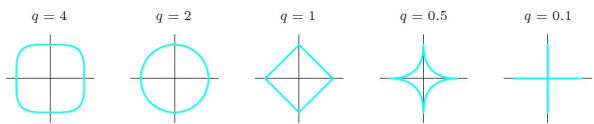


Figure 10: Figure du livre Hastie et al. (2015)

Lasso et Ridge: Comment choisir λ?

- λ est un hyperparamètre, comme d pour la régression polynomiale, ou k dans l'algorithme kNN.
- On sait que la performance sur les données d'entraînement ne fait qu'augmenter si on réduit le biais (petit λ, petit k, grand d).
- Par contre, si on utilise les données de test pour comparer des modèles avec des valeurs différentes de λ, on apprend ce paramètre sur les données de test, qui sont censées servir à l'évaluation...
- On va plutôt utiliser des données de validation: **semaine prochaine**!

Ridge et Lasso: Conclusion

- Régularisation: réduction de variance du modèle au détriment d'un plus grand biais.
- La différence de pénalisation/régularisation entre Lasso et Ridge pousse certains coefficients à 0 avant d'autres pour Lasso. (Le point super important!)
- Lasso nous permet donc de faire de la sélection de variables ET l'estimation de coefficients en simultané pour un λ fixé contrairement aux méthodes exhaustive ou progressive.
- La régularisation Ridge gère bien le problème de colinéarité des variables explicatives.

Pour le prochain cours

- Lecture: 6.1, 6.2 et 6.3 du manuel de référence (James et al. (2013)),
- Lab: 6.5.1 et 6.5.2 (on refait 6.5.2 en gros au lab),
- Exercises 6.6.1, 6.6.3, 6.6.4,
- Préparation : Chapitre 5 du manuel de référence (James et al. (2013)).

Références

Ailliot, P. (2021). Modèles linéaires. EURIA.

Charpentier, A. (2023). *Insurance: biases, discrimination and fairness*. Springer Verlag.

Courville, A. (2024). Representation learning. Université de Montréal.

Friedman, J. H., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC.

James, G., Witten, D., Hastie, T., Tibshirani, R., et al. (2013). *An introduction to statistical learning*, volume 112. Springer.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, USA.

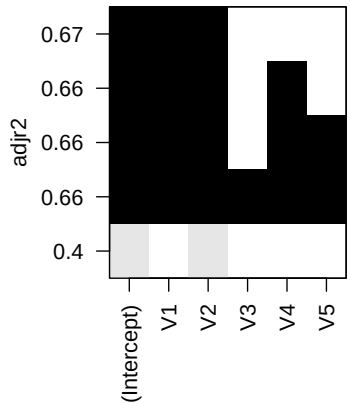
Méthode du meilleur sous-ensemble: En pratique (1/2)

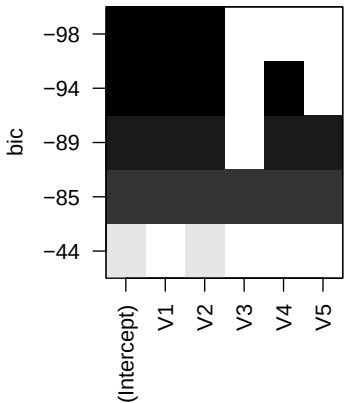
On commence par simuler des données (Ailliot (2021)):

```
> n = 100; p = 5
> X = cbind(rep(1,n), matrix(rnorm(n*p), ncol = p))
> beta = c(rep(1, 3), rep(0, p-2))
> Y = X %*% beta + rnorm(n)
> data = as.data.frame(cbind(Y, X[, -1]))
> colnames(data)[1] <- "Y"
> colnames(data)[2:6] <- paste("V", 1:5, sep = "")
```

Méthode du meilleur sous-ensemble: En pratique (2/2)

```
> library(leaps)
> a <- regsubsets(Y~., nvmax = p, data = data, method = "exhaustive")
> par(mfrow = c(1, 2))
> plot(a, scale = "adjr2"); plot(a, scale = "bic")
```





Méthode *forward*: En pratique

```
> library(MASS)
> fit0 <- lm(Y~1, data = data)
> fit1 <- stepAIC(fit0, scope = list(lower = ~1, upper = ~V1+V2+V3+V4+
    V5), direction = "forward")
> summary(fit1)
Call:
lm(formula = Y ~ V2 + V1, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-2.6020  -0.8425   0.1244   0.6794   2.1947

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.0361     0.1036  10.001  < 2e-16 ***
V2            1.1140     0.1142   9.754 4.56e-16 ***
V1            1.0020     0.1130   8.871 3.67e-14 ***
---
...
```

Méthode *backward*: En pratique

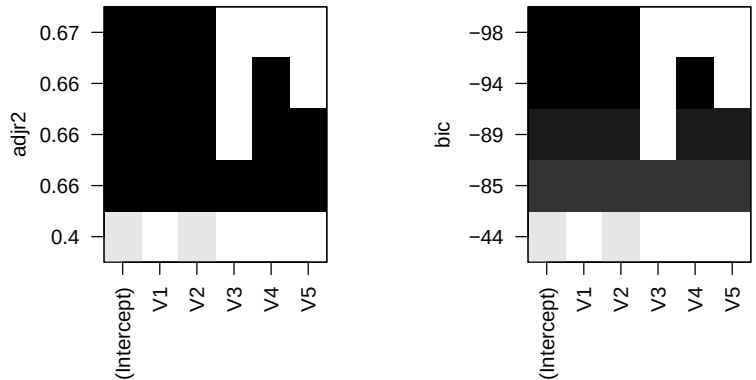
```
> fit <- lm(Y~., data = data)
> fit2 <- stepAIC(fit, direction = "backward")
> summary(fit2)
Call:
lm(formula = Y ~ V1 + V2, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-2.6020 -0.8425  0.1244  0.6794  2.1947

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.0361     0.1036  10.001  < 2e-16 ***
V1            1.0020     0.1130   8.871 3.67e-14 ***
V2            1.1140     0.1142   9.754 4.56e-16 ***
---
...
```

Méthode mixte: En pratique (1/3)

```
> a <- regsubsets(Y~., nvmax = p, data = data, method = "seqrep")
> par(mfrow = c(1, 2))
> plot(a, scale = "adjr2"); plot(a, scale = "bic")
```



Méthode mixte: En pratique (2/3)

```
> fit4 <- stepAIC(fit)
Start:  AIC=15.68
Y ~ V1 + V2 + V3 + V4 + V5
       Df Sum of Sq     RSS     AIC
- V3    1      0.021 103.77 13.699
- V5    1      0.023 103.77 13.701
- V4    1      0.081 103.83 13.757
<none>               103.75 15.679
- V1    1     83.984 187.73 72.984
- V2    1    100.885 204.63 81.604

Step:  AIC=13.7
Y ~ V1 + V2 + V4 + V5
       Df Sum of Sq     RSS     AIC
- V5    1      0.025 103.79 11.724
- V4    1      0.083 103.85 11.779
<none>               103.77 13.699
- V1    1     84.028 187.80 71.019
- V2    1    101.488 205.26 79.909
```

Méthode mixte: En pratique (3/3)

```
    Step:  AIC=11.72
Y ~ V1 + V2 + V4
       Df Sum of Sq    RSS    AIC
- V4    1     0.092 103.89  9.812
<none>              103.79 11.724
- V1    1    84.003 187.80 69.019
- V2    1   101.480 205.27 77.917

Step:  AIC=9.81
Y ~ V1 + V2
       Df Sum of Sq    RSS    AIC
<none>              103.89  9.812
- V1    1    84.281 188.17 67.216
- V2    1   101.891 205.78 76.162
```

Ridge et Lasso: Exemple en dimension 1 ($n = 1$ et $p = 1$)

On suppose un modèle linéaire sans *intercept* β_0. Un seul coefficient doit donc être estimé: $\beta > 0$. On suppose également $x > 0$.

- RSS: $L(\beta) = (y - x\beta)^2 = y^2 - 2xy\beta + x^2\beta^2$,

$$\frac{\partial L(\beta)}{\partial \beta} = 0 \equiv -2xy + 2x^2\beta = 0 \equiv \beta x^2 = xy \equiv \beta = \frac{xy}{x^2}$$

- Ridge: $L(\beta) = (y - x\beta)^2 + \lambda\beta^2 = y^2 - 2xy\beta + x^2\beta^2 + \lambda\beta^2$

$$\frac{\partial L(\beta)}{\partial \beta} = 0 \equiv -2xy + 2(x^2 + \lambda)\beta = 0 \equiv \beta(x^2 + \lambda) = xy \equiv \beta = \frac{xy}{\lambda + x^2}$$

$\beta = 0$ lorsque $\lambda \to +\infty$.

- Lasso: $L(\beta) = (y - x\beta)^2 + \lambda|\beta| = y^2 - 2xy\beta + x^2\beta^2 + \lambda\beta$

$$\frac{\partial L(\beta)}{\partial \beta} = 0 \equiv -2xy + 2x^2\beta + \lambda = 0 \equiv 2\beta x^2 = 2xy - \lambda \equiv \beta = \frac{2xy - \lambda}{2x^2}$$

$\beta = 0$ lorsque $\lambda = 2xy$.

Ridge et Lasso: Exemple avec $n = p$ et $\mathbf{X}$ diagonale (valeurs à 1)

On suppose qu'on performe une régression linéaire sans intercept (James et al. (2013)).

- RSS: $\hat{\beta}_j = y_j$
- Ridge: $\hat{\beta}_j^{\text{Ridge}} = \frac{y_j}{1+\lambda}$
- Lasso: $\hat{\beta}_j^{\text{Lasso}} = \begin{cases} y_j - \frac{\lambda}{2} & \text{si } y_j > \frac{\lambda}{2} \\ y_j + \frac{\lambda}{2} & \text{si } y_j < -\frac{\lambda}{2} \\ 0 & \text{si } |y_j| \leq \frac{\lambda}{2} \end{cases}$

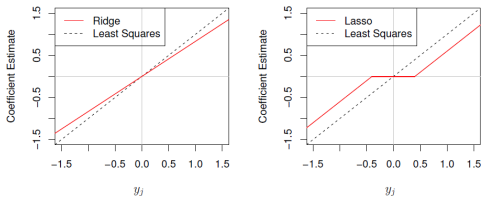


FIGURE 6.10. *The ridge regression and lasso coefficient estimates for a simple setting with $n = p$ and $\mathbf{X}$ a diagonal matrix with 1's on the diagonal.* Left: *The ridge regression coefficient estimates are shrunken proportionally towards zero, relative to the least squares estimates.* Right: *The lasso coefficient estimates are soft-thresholded towards zero.*

Figure 11: Figure 6.10 du livre James et al. (2013)

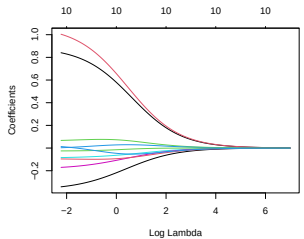
Ridge et Lasso: En pratique (1/5)

Simulation des données (Ailliot (2021)):

```
> n <- 50; p <- 10
> X <- cbind(rep(1,n), matrix(rnorm(n*p),ncol=p))
> beta <- c(rep(1, 3), rep(0, p-2))
> y <- X %*% beta + rnorm(n)
```

Régression Ridge:

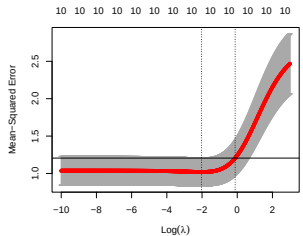
```
> library(glmnet)
> X <- X[, -1]
> fit <- glmnet(X, y,alpha = 0)
> plot(fit, xvar = "lambda")
```



Ridge et Lasso: En pratique (2/5)

Déterminer λ par validation croisée:

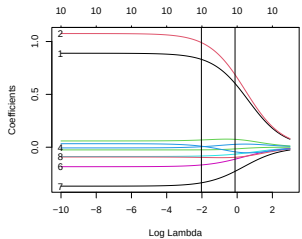
```
> tabl <- exp(seq(-10, 3, by = .01))
> cvfit <- cv.glmnet(X, y, alpha = 0, lambda = tabl, nfolds = length(y))
> plot(cvfit)
> cvfit$lambda.min # valeur optimale de lambda pour MSE
> coef(cvfit, s = "lambda.min") # parametres optimaux
> cvfit$lambda.1se # plus grande valeur de lambda comprise dans 1 standard deviation de la valeur minimale de MSE
> abline(h = cvfit$cvm[cvfit$lambda==cvfit$lambda.1se])
```



Ridge et Lasso: En pratique (3/5)

Déterminer λ par validation croisée:

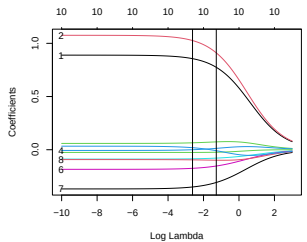
```
> fit <- glmnet(X, y,alpha=0,lambda=tab1)
> plot(fit,xvar="lambda")
> abline(v=log(cvfit$lambda.min)) # lambda avec RMSE minimum
> abline(v=log(cvfit$lambda.1se)) # lambda avec RMSE compris dans RMSE
    minimum plus ecart-type de l'erreur
> vnat <- coef(fit)
> vnat <- vnat[-1,ncol(vnat)]
> axis(2, at = vnat, line = -2, label = as.character(1:10), las = 1,
   tick = FALSE, cex.axis = 1)
```



Ridge et Lasso: En pratique (4/5)

Régression Lasso:

```
> tabl <- exp(seq(-10, 1, by = .01))
> cvfit <- cv.glmnet(X, y, alpha = 1, lambda = exp(seq(-10, 1, by =
    .01)), nfolds = length(y))
> plot(fit, xvar = "lambda")
> abline(v = log(cvfit$lambda.min))
> abline(v = log(cvfit$lambda.1se))
> vnat <- coef(fit)
> vnat <- vnat[-1,ncol(vnat)]
> axis(2, at = vnat, line = -2, label = as.character(1:10), las = 1,
    tick = FALSE, cex.axis = 1, col = 'red')
```



Ridge et Lasso: En pratique (5/5)

```
> coef(cvfit, s = "lambda.min")
> coef(cvfit, s = "lambda.1se")
> plot(cvfit)
```