Validation Croisée

Joseph Salmon, Nicolas Verzelen

Université de Montpellier / INRA

Évaluer la précision d'une règle d'apprentissage

Supposons que l'on ajuste une règle $\widehat{f}(x)$ sur les données $D_1^n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$.

Le **risque empirique** ou **risque apparent** d'un algorithme de $\widehat{f}$ (construit sur D_1^n) est défini par $\widehat{R}_n(\widehat{f}) = \frac{1}{n} \sum_{i=1}^{n} l\left(y_i, \widehat{f}(x_i)\right)$.

ATTENTION : Sous-estimation du risque moyen

$\Rightarrow$ Minimisation du risque empirique = sur-apprentissage (overfitting) !

Erreur de *Test*

Supposons que l'on a accès à un deuxième échantillon

$$D_{n+1}^{n+N} = \{(x_{n+1}, y_{n+1}), \ldots, (x_{n+N}, y_{n+N})\}$$

indépendant de l'échantillon d'apprentissage D_1^n.

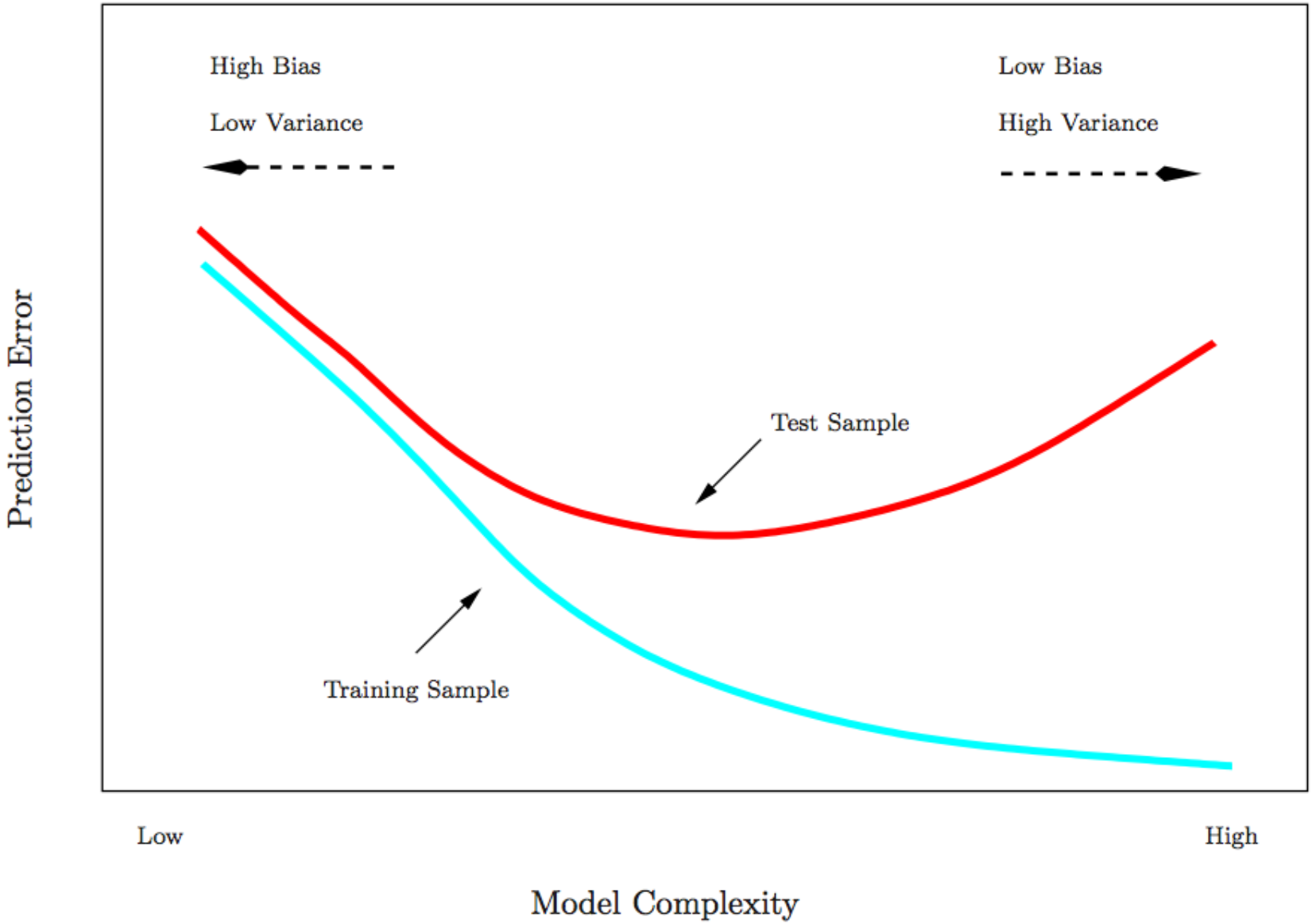
$$\widetilde{R}_{Te}(\widehat{f}) = \frac{1}{N} \sum_{i=n+1}^{N+n} l(y_i, \widehat{f}(x_i))$$

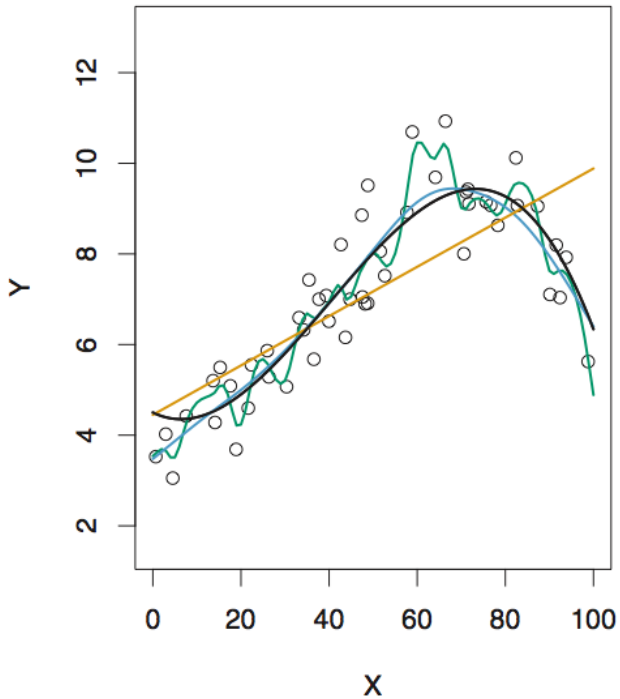
Proposition

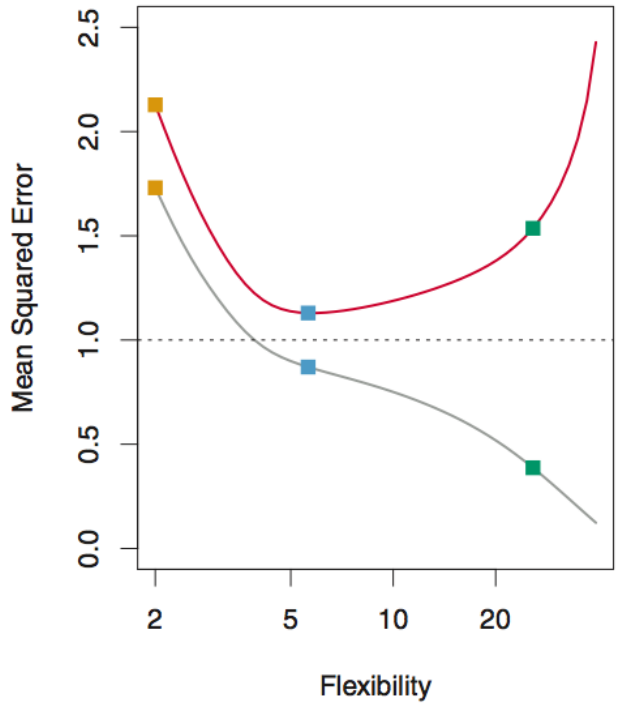
$$\mathbb{E}[\widetilde{R}_{Te}(\widehat{f})|D_1^n] = R_P(\widehat{f})$$

Lorsque N tend l'infini, $\widetilde{R}_{Te}(\widehat{f})$ converge presque surement vers $R_P(\widehat{f})$.

Erreur d'entrainement et erreur de test

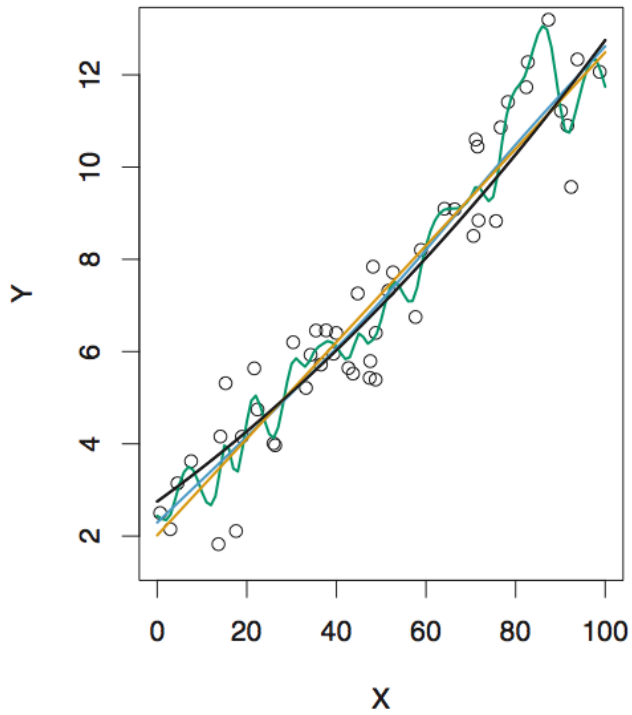


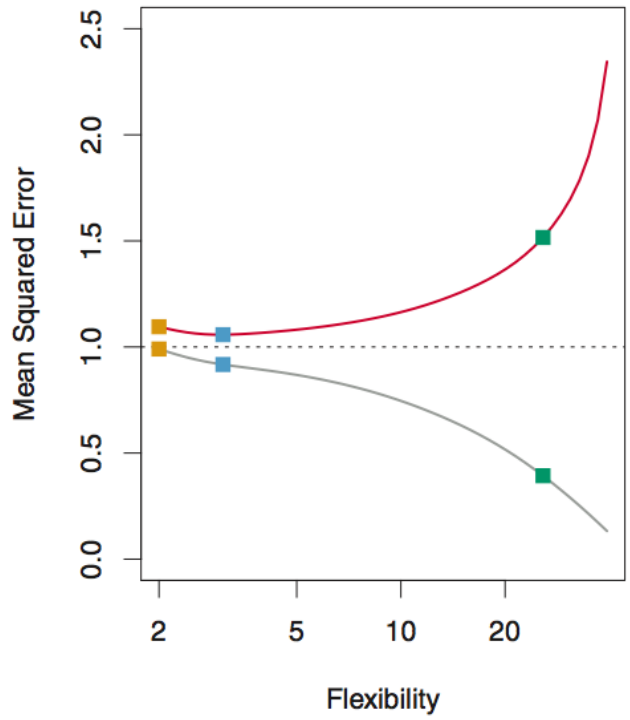




Noire : vraie $f(x)$
Orange : modèle linéaire
Bleue et verte : splines

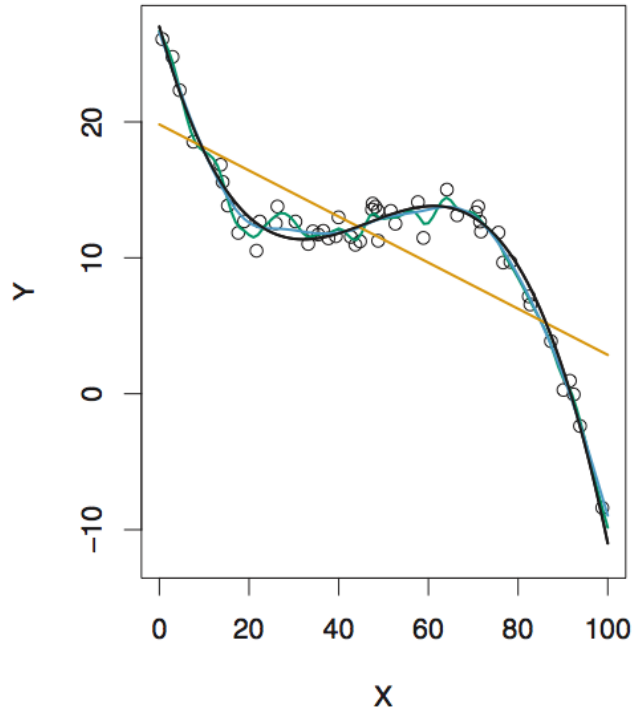
Rouge : risque test $\widetilde{R}_{Te}(\widehat{f})$
Gris : risque empirique $\widehat{R}_n(\widehat{f})$

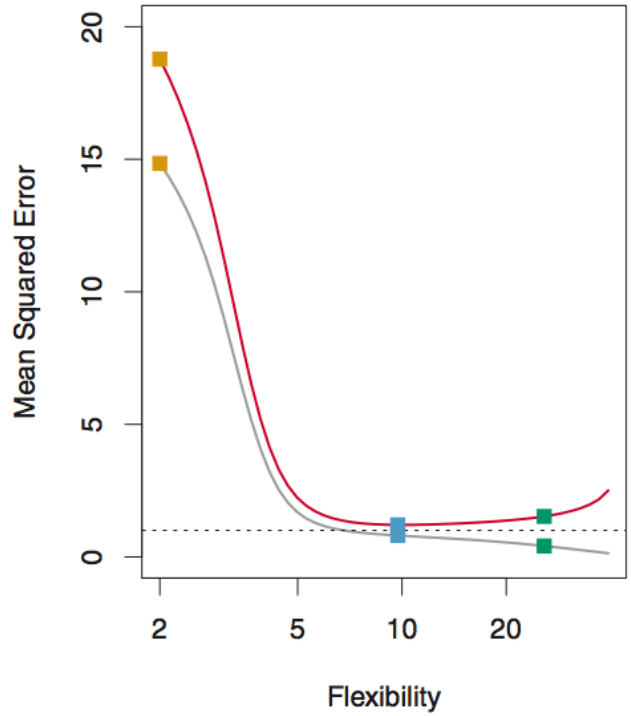




Noire : vraie $f(x)$
Orange : modèle linéaire
Bleue et verte : splines

Rouge : risque test $\widetilde{R}_{Te}(\widehat{f})$
Gris : risque empirique $\widehat{R}_n(\widehat{f})$





Noire : vraie $f(x)$
Orange : modèle linéaire
Bleue et verte : splines

Rouge : risque test $\widetilde{R}_{Te}(\widehat{f})$
Gris : risque empirique $\widehat{R}_n(\widehat{f})$

Estimations de l'erreur de prédiction

- La meilleure solution : un grand ensemble de test clairement désigné. Bien souvent, ce n'est pas disponible.

- Certaines méthodes permettent de corriger l'erreur d'entrainement pour estimer l'erreur de test, avec des arguments fondés mathématiquement.

  Cela inclut les critères AIC et BIC. Ils seront discutés plus tard.

- Ici, nous nous intéressons à une classe de méthodes qui estime le risque en mettant de côté un sous-ensemble des données d'entrainement disponibles pour ajuster les modèles, et en appliquant la méthodes ajustée sur ces données mises de côté.

Approche par ensemble de validation

- Cette méthode propose de diviser l'échantillon d'apprentissage en deux : un ensemble d'entrainement et un ensemble de validation
- Le modèle est ajusté sur l'ensemble d'entrainement, et on l'utilise ensuite pour prédire les réponses sur l'échantillon de validation.

Approche par ensemble de validation

Algorithme : Algorithme de validation croisée hold-out

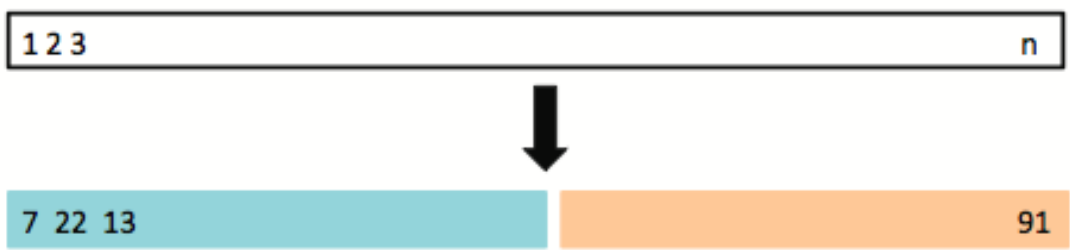
input : $\mathcal{A}$ sous-ensemble de $\{1, \ldots, n\}$, définissant l'ensemble d'apprentissage

input : $\mathcal{V}$ sous-ensemble de $\{1, \ldots, n\}$, définissant l'ensemble de validation

début

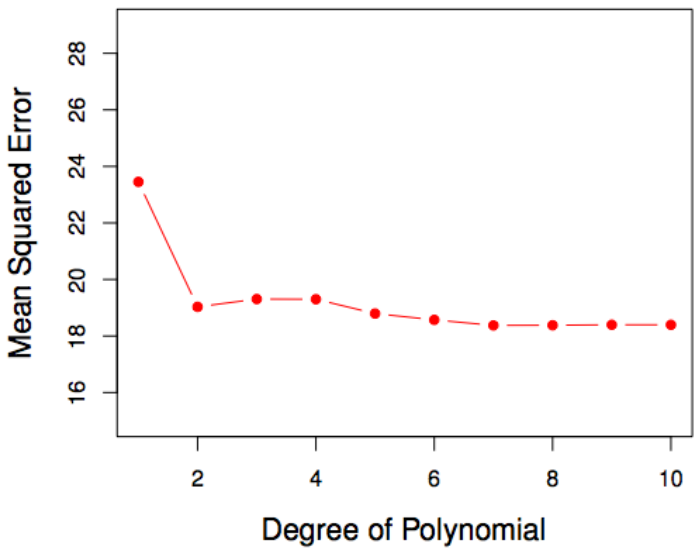
Construire la règle de prédiction $\widehat{f}_{D_{\mathcal{A}}}$ sur $D_{\mathcal{A}} = \{(x_i, y_i),\ i \in \mathcal{A}\}$

output : $\frac{1}{\#\mathcal{V}} \sum_{i \in \mathcal{V}} l\left(y_i, \widehat{f}_{D_{\mathcal{A}}}(x_i)\right)$

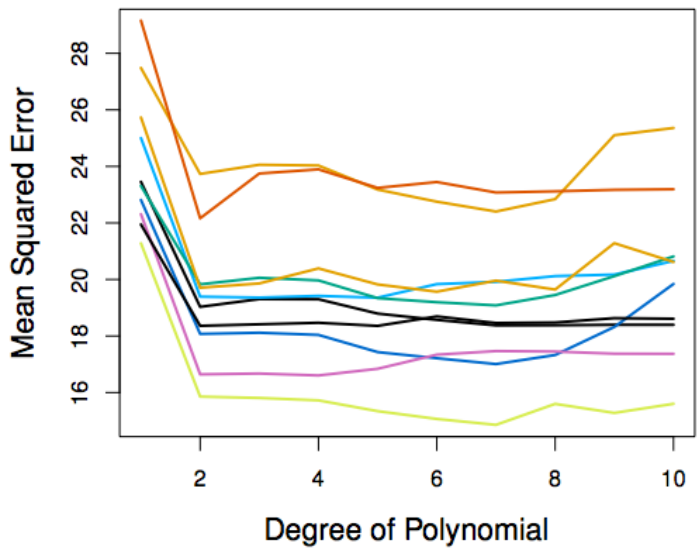


Exemple sur les données automobiles

- On veut comparer la régression linéaires à des régressions polynomiales de différents degrés
- On divise en deux les 392 observations : 196 pour l'entrainement, 196 pour le test.



Sur une partition aléatoire



Variabilité d'une partition à l'autre

Inconvénients de l'approche par ensemble de validation

- L'estimation obtenue par cette méthode peut être très variable, et dépend de la chance ou malchance dans la construction du sous-'échantillon de validation
- Dans cette approche, seule une moitié des observations est utilisée pour ajuster les modèles — celles qui sont dans l'ensemble d'entrainement.
- Cela suggère que l'erreur calculée peut surestimer l'erreur de test d'un modèle ajusté sur l'ensemble des données (moins de variabilité d'échantillonnage dans l'inférence des paramètres du modèle)

Déjà mieux : échanger les rôles entrainement-validation et faire la moyenne des deux erreurs obtenues. On *croise* les rôles.

Estimation du risque moyen

Méthodes de validation croisée et de bootstrap

- Validation croisée leave-p-out
- Validation croisée K fold.

Algorithme : Algorithme de validation croisée leave p out

input : p entier inférieur à n

début

Construire $\mathcal{V}_1, \ldots, \mathcal{V}_{\binom{n}{p}}$ parties à p éléments de $\{1, \ldots, n\}$

pour $k = 1, \ldots, \binom{n}{p}$ **faire**

Déterminer $\mathcal{A}_k = \{1, \ldots, n\} \setminus \mathcal{V}_k$

Construire $\widehat{f}_{D_{\mathcal{A}_k}}$ sur $D_{\mathcal{A}_k} = \{(x_i, y_i),\ i \in \mathcal{A}_k\}$

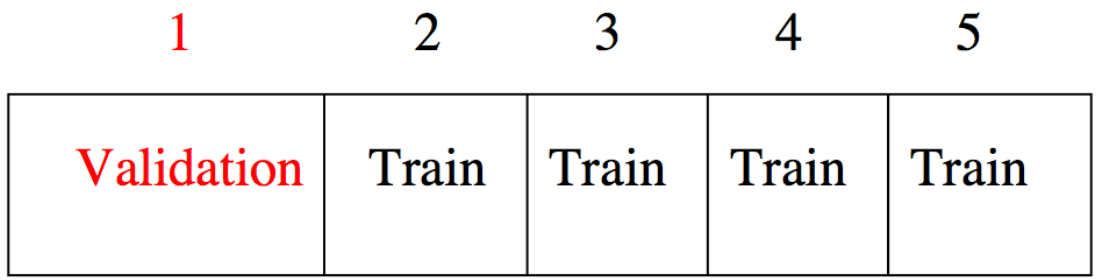
Calculer $R_k = \frac{1}{p} \sum_{i \in \mathcal{V}_k} l\left(y_i, \widehat{f}_{D_{\mathcal{A}_k}}(x_i)\right)$

output : $\binom{n}{p}^{-1} \sum_{k=1}^{\binom{n}{p}} R_k$

Rem: : Temps de calcul très long hormis pour $p = 1$ (ou $p = 2$)

Validation croisée à K groupes

- C'est la méthode la plus couramment utilisée pour estimer l'erreur de test
- L'estimation peut être utilisée pour choisir le meilleur modèle (la meilleure méthode d'apprentissage), ou approcher l'erreur de prédiction du modèle finalement choisi.
- L'idée est de diviser les données en K groupes de même taille. On laisse le k-ème bloc de côté, on ajuste le modèle, et on l'évalue sur le bloc laissé de côté.
- On répète l'opération en laissant de côté le bloc $k = 1$, puis $k = 2, \ldots$ jusqu'à $k = K$. Et on combine les résultats



Algorithme : Algorithme de validation croisée K-fold

input : K entier diviseur de n

début

Construire $\mathcal{V}_1, \ldots, \mathcal{V}_K$ partition de $\{1, \ldots, n\}$ **pour** $k = 1, \ldots, K$ **faire**

Déterminer $\mathcal{A}_k = \{1, \ldots, n\} \setminus \mathcal{V}_k$

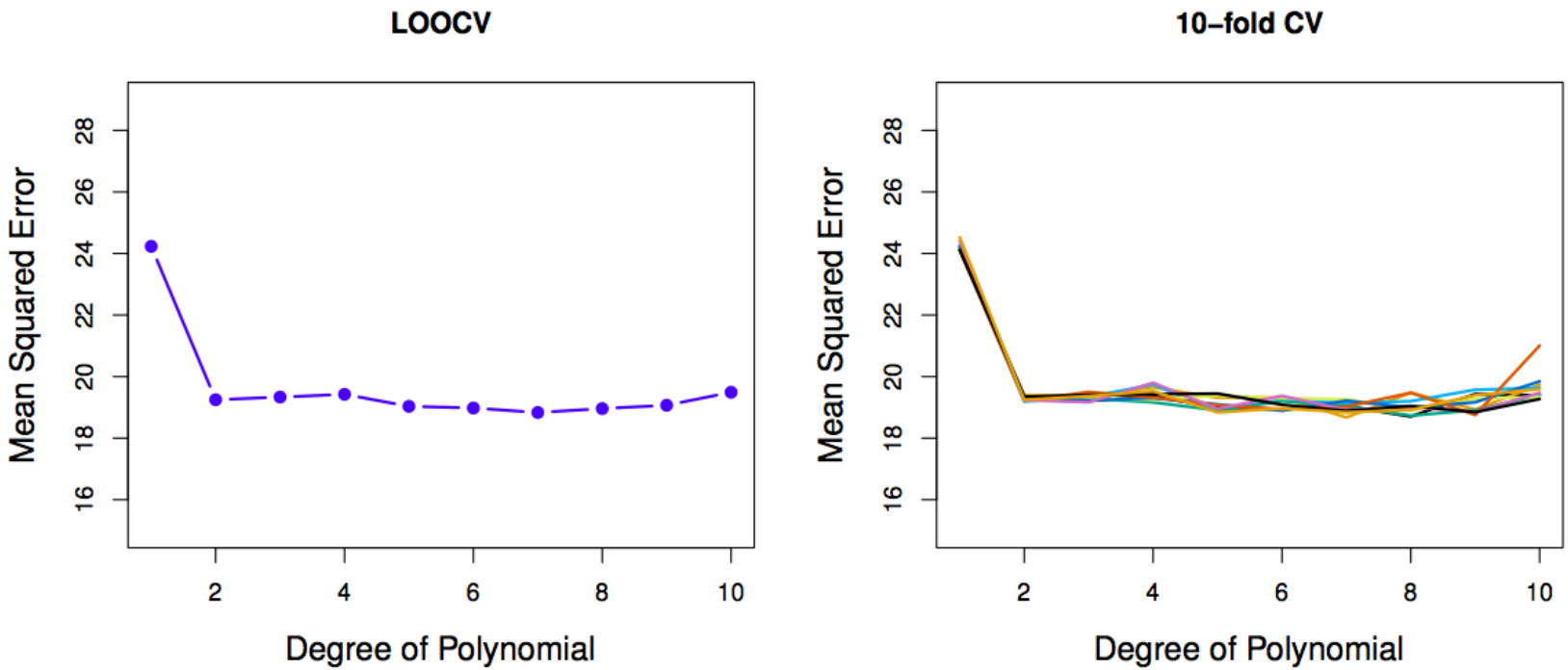
Construire $\widehat{f}_{D_{\mathcal{A}_k}}$ sur $D_{\mathcal{A}_k} = \{(x_i, y_i),\ i \in \mathcal{A}_k\}$

Calculer $R_k = \frac{K}{n} \sum_{i \in \mathcal{V}_k} l\left(y_i, \widehat{f}_{D_{\mathcal{A}_k}}(x_i)\right)$

output : $\frac{1}{K} \sum_{k=1}^{K} R_k$

- On choisit généralement $K = 5$ ou $K = 10$ blocs
- Lorsque $K = n$, il s'agit de la méthode de « *leave-one ou cross-validation* » (LOOCV)

Retour sur les données automobile



En cas d'égalité, choisir le modèle le plus *parcimonieux* car il aura naturellement moins de variance d'estimation dans les coefficients du modèle.

Validation croisée : les pièges

Considérons un classifieur simple pour prédire une réponse Y binaire, mais avec de nombreuses co-variables ($p = 5000$).

On procède comme suit

1. On démarre avec les 5000 prédicteurs et un échantillon de taille 50, et on cherche les 100 prédicteurs qui ont la plus grande corrélation par la réponse
2. On utilise une méthode d'apprentissage, par exemple la régression logistique, sur ces 100 meilleures co-variables.

Comment doit-on estimer l'erreur de test ?

Peut-on utiliser la validation croisée à l'étape 2 ? ? ?

Réponse : NON, NON et NON !

1. On démarre avec les 5000 prédicteurs et un échantillon de taille 50, et on cherche les 100 prédicteurs qui ont la plus grande corrélation par la réponse
2. On utilise une méthode d'apprentissage, par exemple la régression logistique, sur ces 100 meilleures co-variables.

- Cela ignore le fait que l'étape 1 a déjà utilisé les réponses observées. Cette étape est une forme d'entrainement du classifier final.
- Il est facile de simuler des données réalistes, avec Y indépendant de X (dont l'erreur de test doit être de 50%) mais dont l'erreur calculée par CV dans l'étape 2 est proche de 0 !
  Essayer de le faire vous même !
- Cette erreur est pourtant comise dans de nombreux articles traitant de données génomiques !

Pré-validation

- Dans des études génomiques (microarray, etc.), un problème important est de comparer un prédicteur de l'état d'une maladie, calculé à partir des données microarray, avec d'autre prédicteurs cliniques
- Ce prédicteur est un résumé numérique construit sur des données. Comparer son lien avec l'état de la maladie sur les données qui ont servi à le construire n'est pas juste envers les autres prédicteurs
- La pré-validation permet de dé-biaiser la situation.

Exemple. van't Veer et al. *Nature* (2002)

Données microarray de 4918 gènes sur 78 cas de cancer des poumons : 44 cas favorables, 34 cas graves.

On construit le prédicteur (un résumé binaire) $z_i = \hat{C}(x_i)$ pour chacun des cas i de l'échantillon.

On veut maintenant comparer ce résumé de données génomiques avec d'autres variables pour prédire l'état d'un nouveau patient.

Régression logistique sans pré-validation

	coeff	Stand.Err	Z-stat	p-value
microarray $\hat{C}$	4.096	1.092	3.752	$<$0.001
angio	1.208	0.816	1.482	0.069
er	-0.554	1.044	-0.530	0.298
grade	-0.697	1.003	-0.695	0.243
pr	1.214	1.057	1.149	0.125
age	-1.593	0.911	-1.748	0.040
size	1.483	0.732	2.026	0.021

$\hat{C}$ est déjà lié au données (par construction de $\hat{C}$). Ce n'est pas étonnant qu'il est une p-value très faible.

Cela ne dit rien de la capacité de $\hat{C}$ sur de nouveaux patients.

Solution : pré-validation

On procède comme suit.

1. On divise les cas observés en $K = 13$ groupes de 6 observations chacun.
2. On construit un $\hat{C}$ sur 12 groupes.
3. On l'utilise pour construire la co-variable sur les données mises de côté, ce qui donne des $\tilde{Z}_i$ pour le groupe mis de côté
4. On répète les étapes 2 et 3 pour construire les $\tilde{Z}_i$ sur chacune des observations
5. On utilise ce prédicteur, pour le comparer aux 6 autres prédicteurs cliniques

	coeff	Stand.Err	Z-stat	p-value
microarray $\tilde{Z}$	1.549	0.675	2.296	0.011
angio	1.589	0.682	2.329	0.010
er	-0.617	0.894	-0.690	0.245
grade	0.719	0.720	0.999	0.159
pr	0.537	0.863	0.622	0.267
age	-1.471	0.701	-2.099	0.018
size	0.998	0.594	1.681	0.046