

Séance 5 : Sélection - Validation de modèles

Ibrahim TOURE,
Ingénieur Statisticien

Université d'Evry Val d'Essone

14 Janvier 2017

Sommaire

- 1 Sélection-Choix de modèles
- 2 Sélection de variables
- 3 Validation de modèles
- 4 Examen des résidus

Sélection-Choix de modèles

Le problème de **sélection/choix de modèles** est ici abordé sous 2 angles (différents mais pas nécessairement indépendants) :

- ① On est en présence de M_1, \dots, M_k modèles et on se pose le problème d'en choisir un.
 - Il n'existe pas de **critère universel** permettant de définir la notion de meilleur modèle.
 - On parlera toujours de meilleur modèle **par rapport à un critère donné**.
 - On présentera deux types de critère :
 - **ajustement du modèle** (vraisemblances pénalisées)
 - **capacité de prédiction du modèle**
- ② Etant donnés Y une variable à expliquer et X_1, \dots, X_p p variables explicatives, comment sélectionner **automatiquement** un modèle logistique ?

Il s'agit de trouver automatiquement un sous-groupe des variables explicatives permettant d'expliquer Y . On parle de **sélection de variables**.

Tests entre modèles emboîtés

- Afin de simplifier les notations, on supposera que l'on est en présence de deux modèles candidats \mathcal{M}_1 et \mathcal{M}_2 .
- Nous nous plaçons dans le cas particulier où le modèle \mathcal{M}_1 est emboîté dans \mathcal{M}_2 (\mathcal{M}_1 est un cas particulier de \mathcal{M}_2).

Exemple

\mathcal{M}_1 et \mathcal{M}_2 sont respectivement définis par

$$\text{logit } p_{\beta}(x) = \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$$

et

$$\text{logit } p_{\beta}(x) = \beta_1 x_1 + \beta_2 x_2.$$

Un moyen naturel de comparer \mathcal{M}_1 et \mathcal{M}_2 consiste à tester

$$H_0 : \beta_3 = \beta_4 = 0 \quad \text{contre} \quad H_1 : \beta_3 \neq 0 \text{ ou } \beta_4 \neq 0.$$

Tests entre modèles emboîtés

- Plus généralement, considérons \mathcal{M}_1 et \mathcal{M}_2 deux modèles logistiques à p_1 et p_2 paramètres tels que \mathcal{M}_1 est emboité dans \mathcal{M}_2 .
- Tester \mathcal{M}_1 contre \mathcal{M}_2 revient à **tester la nullité des coefficients** de \mathcal{M}_2 que ne sont pas dans \mathcal{M}_1 .
- On sait faire... On peut mettre en oeuvre un test de **Wald, du rapport de vraisemblance ou du score**.
- Sous H_0 ces 3 statistiques de test suivent une loi du $\chi^2_{p_2-p_1}$.

Exemple

- Pour le problème sur la maladie cardiovasculaire, on souhaite **comparer les modèles**

```
> model1 <- glm(chd~tobacco+famhist,data=SAheart,family=binomial)
> model2 <- glm(chd~tobacco+famhist+adiposity+alcohol,data=SAheart,
               family=binomial)
```

à l'aide d'un test de rapport de vraisemblance.

- On peut calculer la probabilité critique **à la main**

```
> stat <- 2*(logLik(model2)-logLik(model1))
> stat[1]
[1] 11.11016
> 1-pchisq(stat,df=length(model2$coef)-length(model1$coef))
[1] 0.003867751
```

- Ou directement avec la fonction **anova**

```
> anova(model1,model2,test="LRT")
Analysis of Deviance Table

Model 1: chd ~ tobacco + famhist
Model 2: chd ~ tobacco + famhist + adiposity + alcohol
    Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1         459      524.58          0.000000
2         457      513.47   2      11.11 0.003868 **
```

Figure :

AIC-BIC

- **Idée** : utiliser la vraisemblance pour comparer \mathcal{M}_1 et \mathcal{M}_2 .

Problème

Si $\mathcal{M}_1 \subset \mathcal{M}_2$ alors $\mathcal{L}_n(\hat{\beta}_1) \leq \mathcal{L}_n(\hat{\beta}_2)$ où $\hat{\beta}_j$ désigne l'emv du modèle $\mathcal{M}_j, j = 1, 2$.

- **Conséquence** : la vraisemblance sélectionnera toujours le modèle le plus complexe.

Solution

Pénaliser la vraisemblance par la complexité du modèle.

AIC-BIC

Définition

Soit \mathcal{M} un modèle logistique à p paramètres. On note $\hat{\beta}_n$ l'emv des paramètres du modèle.

- L'**AIC (Akaike Information Criterion)** du modèle \mathcal{M} est défini par

$$AIC(\mathcal{M}) = -2\mathcal{L}_n(\hat{\beta}_n) + 2p.$$

- Le **BIC (Bayesian Information Criterion)** du modèle \mathcal{M} est défini par

$$BIC(\mathcal{M}) = -2\mathcal{L}_n(\hat{\beta}_n) + p \log n.$$

- Le modèle retenu sera celui qui **minimise** l'AIC ou le BIC.
- $\log n > 2$ (pour $n \geq 8$) BIC aura tendance à choisir des modèles plus **parcimonieux** que AIC.

Exemple

- On considère les mêmes modèles que précédemment :

```
> model1 <- glm(chd~tobacco+famhist,data=SAheart,family=binomial)
> model2 <- glm(chd~tobacco+famhist+adiposity+alcohol,data=SAheart,
                family=binomial)
```

- On les compare en terme d'AIC et de BIC.

```
> c(AIC(model1),AIC(model2))
[1] 530.5759 523.4657
> c(BIC(model1),BIC(model2))
[1] 542.9826 544.1436
```

Conclusion

AIC sélectionne `model2` tandis que BIC sélectionne `model1`.

Figure :

Prévisions

- L'idée est de chercher à comparer les **pouvoirs de prédiction** des modèles concurrents et de choisir celui qui prédit le mieux.
- L'approche consiste à définir une **règle de classification** à partir d'un modèle logistique :

$$\hat{g} : \mathbb{R}^p \rightarrow \{0, 1\}$$

qui à une valeur observée des variables explicatives associe une valeur prédicte pour Y .

- Il existe plusieurs critères permettant de **mesurer la performance** d'une règle \hat{g} .
- Un des critères les plus classiques consiste à chercher à estimer la **probabilité d'erreur**

$$P(\hat{g}(X) \neq Y).$$

Prévision avec un modèle logistique

- Modèle logistique permettant d'expliquer Y par X_1, \dots, X_p :

$$\text{logit } p_{\beta}(x) = \beta_1 x_1 + \dots + \beta_p x_p = x' \beta.$$

- On peut estimer $p_{\beta}(x)$ par

$$p_{\hat{\beta}_n}(x) = \frac{\exp(x' \hat{\beta}_n)}{1 + \exp(x' \hat{\beta}_n)}.$$

- Un moyen naturel de prédire le label y_{n+1} d'un nouvel individu x_{n+1} est de poser

$$\hat{Y}_{n+1} = \begin{cases} 1 & \text{si } p_{\hat{\beta}_n}(x) \geq s \\ 0 & \text{sinon.} \end{cases}$$

Remarque

Le seuil s doit être choisi par l'utilisateur. Les logiciels prennent souvent par défaut $s = 0.5$.

Intervalles de confiance pour la probabilité estimée

- Etant donnée un nouvel individu x_{n+1} , il peut être intéressant de construire un **intervalle de confiance** pour la probabilité $p_\beta(x_{n+1})$.
- $\hat{\beta}_n$ est (pour n grand) un **vecteur gaussien** d'espérance β et de matrice de variance-covariance $\mathcal{I}_n(\beta)^{-1}$.
- Par conséquent, $x'_{n+1}\hat{\beta}_n$ est (pour n grand) une var de loi gaussienne

$$\mathcal{N}(x'_{n+1}\beta, x'_{n+1}\mathcal{I}_n(\beta)^{-1}x_{n+1}).$$

En posant $\hat{\sigma}^2 = x'_{n+1}(\mathbb{X}'W_{\hat{\beta}}\mathbb{X})^{-1}x_{n+1}$, on déduit

$$IC_{1-\alpha}(p_\beta(x_{n+1})) = \left[\frac{\exp(x'_{n+1}\hat{\beta}_n - u_{1-\alpha/2}\hat{\sigma})}{1 + \exp(x'_{n+1}\hat{\beta}_n - u_{1-\alpha/2}\hat{\sigma})}; \frac{\exp(x'_{n+1}\hat{\beta}_n + u_{1-\alpha/2}\hat{\sigma})}{1 + \exp(x'_{n+1}\hat{\beta}_n + u_{1-\alpha/2}\hat{\sigma})} \right]$$

- **Remarque** : il est possible de construire un IC centré en utilisant la delta-méthode.

Un critère de prévision : la probabilité d'erreur

- On suppose dans cette section que les **variables explicatives sont aléatoires** et on note $\mathcal{D}_n = (X_1, Y_1), \dots, (X_n, Y_n)$ un n -échantillon i.i.d de même loi que (X, Y) .
- L'approche consiste à comparer deux modèles \mathcal{M}_1 et \mathcal{M}_2 en **comparant les probabilités d'erreur**

$$L(\hat{g}) = \mathbf{P}(\hat{g}(X) \neq Y)$$

des règles de classification issues de ces deux modèles.

La probabilité $L(\hat{g})$ est **inconnue** et doit être **estimée**.

Figure :

Un critère de prévision : la probabilité d'erreur

- **Première idée** : estimer $L(\hat{g})$ en
 - ① **appliquant la règle \hat{g}** sur les variables X_i pour en déduire une **prévision** $\hat{Y}_i = \hat{g}(X_i)$ de la variable Y pour chaque individu.
 - ② comparant la **prévision** \hat{Y}_i avec la **valeur observée** Y_i

$$L_n(\hat{g}) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}_{\hat{g}(X_i) \neq Y_i}.$$

Table de confusion

- On dresse généralement la table de confusion

	$\hat{Y} = 0$	$\hat{Y} = 1$
$Y = 0$	OK	E_1
$Y = 1$	E_2	OK

- La probabilité d'erreur est alors **estimée** par

$$L_n(\hat{g}) = \frac{E_1 + E_2}{n}.$$

Un critère de prévision : la probabilité d'erreur

Problème

- $L_n(\hat{g})$ n'est généralement **pas un bon estimateur** de $L(\hat{g})$ (sous-estimation).
- La loi des grands nombres ne peut s'appliquer car les variables

$$\mathbf{1}_{\hat{g}(X_i) \neq Y_i}$$

ne sont **pas indépendantes**.

- Le problème vient du fait que l'échantillon \mathcal{D}_n est **utilisé deux fois** (pour calculer \hat{g} puis pour estimer $L(\hat{g})$).

Solution

Découper l'échantillon en deux :

- **un échantillon d'apprentissage** utilisé pour calculer la règle \hat{g} (estimer les paramètres du modèle logistique).
- **un échantillon test ou de validation** utilisé pour estimer la probabilité d'erreur $L(\hat{g})$.

Procédure A/V

Estimateur de $L(\hat{g})$ par A/V

L'échantillon $\mathcal{D}_n = (X_1, Y_1), \dots, (X_n, Y_n)$ est séparé aléatoirement en deux sous échantillons :

- ① un échantillon d'apprentissage $\mathcal{D}_\ell = \{(X_i, Y_i), i \in \mathcal{J}_\ell\}$ de taille ℓ utilisé pour estimer les paramètres du modèle et en déduire la règle \hat{g} .
- ② un échantillon test ou de validation $\mathcal{D}_m = \{(X_i, Y_i), i \in \mathcal{J}_m\}$ de taille m utilisé pour estimer $L(\hat{g})$ par

$$L_n(\hat{g}) = \frac{1}{m} \sum_{i \in \mathcal{J}_m} \mathbf{1}_{\hat{g}(X_i) \neq Y_i},$$

avec $\mathcal{J}_\ell \cup \mathcal{J}_m = \{1, \dots, n\}$ et $\mathcal{J}_\ell \cap \mathcal{J}_m = \emptyset$.

Propriété

L'estimateur $L_n(\hat{g})$ est un estimateur **sans biais** de $L(\hat{g})$.

Exemple

On estime la probabilité d'erreur pour deux modèles logistiques concurrents sur les données concernant la maladie cardiovasculaire.

- Construction des échantillons d'apprentissage et test

```
n <- nrow(SAheart)
l <- 250 #taille de l'ech d'apprentissage
set.seed(1234)
perm <- sample(n)
dapp <- SAheart[perm[1:l],]
dtest <- SAheart[-perm[1:l],]
```

- Ajustement des modèles sur l'échantillon d'apprentissage.

```
> model1 <- glm(chd~tobacco+famhist,data=dapp,family=binomial)
> model2 <- glm(chd~tobacco+famhist+adiposity+alcohol,data=dapp,famil
```

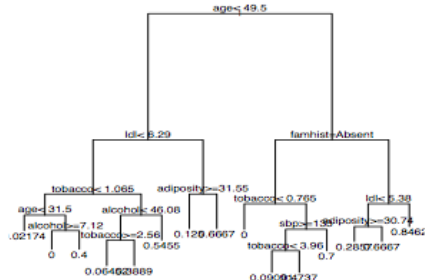
- Estimation de la probabilité d'erreur sur l'échantillon test.

```
> prev1 <- round(predict(model1,newdata=dtest,type="response"))
> prev2 <- round(predict(model2,newdata=dtest,type="response"))
>
> mean(prev1!=dtest$chd)
[1] 0.3113208
> mean(prev2!=dtest$chd)
[1] 0.2877358
```

Exemple

- Un des avantages de la probabilité d'erreur est qu'elle permet de comparer **différents modèles issus de différentes méthodes**.
- Construisons par exemple un **arbre de classification**.

```
> arbre <- rpart(chd~.,data=dapp)
> plot(arbre)
> text(arbre,pretty=0)
```



Exemple

- Et estimons sa **probabilité d'erreur** sur l'échantillon test

```
> prev3 <- predict(arbre, newdata=dtest, type="class")  
> mean(prev3!=dtest$chd)  
[1] 0.3490566
```

Probabilités d'erreur estimées

modèle	logit1	logit2	arbre
erreur estimée	0.31	0.29	0.35

- Pour ce critère, on privilégiera le **second modèle logistique**.

Un inconvénient de la probabilité d'erreur

- Le critère de la probabilité d'erreur porte sur une **règle de classification**.
- Il **impose donc d'avoir fixé le seuil s** tel que

$$\hat{Y} = \begin{cases} 1 & \text{si } p_{\hat{\beta}_n}(x) \geq s \\ 0 & \text{sinon.} \end{cases}$$

et il dépend du seuil fixé.

- Il existe des **indicateurs plus flexibles** (toujours basés sur la prévision) qui n'imposent pas de fixer le seuil.
- La **courbe ROC** basée sur la notion de **score** fait partie de ces critères.

Score : définition

- Un score est une **fonction** $S : \mathbb{R}^p \rightarrow \mathbb{R}$.
- Etant données n un échantillon $(X_1, Y_1), \dots, (X_n, Y_n)$ le job du statisticien consiste à **construire une fonction** $S(x)$ qui permettent d'expliquer Y *au mieux*.



- Une fois le score construit, la **décision** s'effectue selon la procédure

$$\hat{Y} = \begin{cases} 1 & \text{si } \hat{S}(x) \geq s \\ 0 & \text{sinon,} \end{cases}$$

où s est un **seuil** choisi par l'utilisateur.

- La construction de scores s'effectue généralement avec les **modèles de régression classiques**.

Score logistique

- de loin le plus utilisé...
- On considère le **modèle logistique**

$$\log \frac{p_{\beta}(x)}{1 - p_{\beta}(x)} = \beta_1 x_1 + \dots + \beta_p x_p$$

- Il suffit de poser $S(x) = p_{\beta}(x)$ ou

$$S(x) = \beta_1 x_1 + \dots + \beta_p x_p.$$

Illustration

- On calcule un **score logistique** sur l'exemple suivant :
- On dispose d'un **échantillon de taille** $n = 150$ pour construire les fonctions de score (table dapp) :

	X1	X2	Y
1	-1.2070657	0.3158544	1
2	0.2774292	-2.1866448	0
3	1.0844412	-0.3307386	0
4	-2.3456977	-1.9001806	1
5	0.4291247	-0.3691092	0

- On souhaite calculer le score pour 100 **nouveaux individus** (table dtest) :

	X1	X2
151	-0.37723765	-0.01545427
152	0.09761946	1.65997581
153	1.63874465	1.24334905
154	-0.87559247	-0.00564424
155	0.12176000	0.44504449

Illustration

- On ajuste le **modèle logistique** sur l'échantillon d'apprentissage :

```
> model_logit <- glm(Y~.,data=dapp,family=binomial)
```

- On calcule le score des **nouveaux individus** :

```
> S1 <- predict(model_logit,newdata=dtest,type="response")
```

- On peut afficher le score de ces nouveaux individus :

```
> S1[1:5]
      151      152      153      154      155
0.77724343 0.56927363 0.02486394 0.92479413 0.51310825
```


Table de confusion

- Etant donnés un score et un seuil s , on peut se donner une règle de décision

$$\hat{Y} = \hat{Y}_s = \begin{cases} 1 & \text{si } S(x) \geq s \\ 0 & \text{sinon.} \end{cases}$$

- Cette règle définit la table de confusion

	$\hat{Y} = 0$	$\hat{Y} = 1$
$Y = 0$	OK	E_1
$Y = 1$	E_2	OK

- On définit :

- **Spécificité** : $sp(s) = \mathbf{P}(S(X) < s | Y = 0)$
- **Sensibilité** : $se(s) = \mathbf{P}(S(X) \geq s | Y = 1)$

Courbe ROC

Définition

C'est une courbe paramétrée par le seuil :

$$\begin{cases} x(s) = 1 - sp(s) = \mathbf{P}(S(X) > s | Y = 0) \\ y(s) = se(s) = \mathbf{P}(S(X) \geq s | Y = 1) \end{cases}$$

Figure :

Courbe ROC

Définition

C'est une courbe paramétrée par le seuil :

$$\begin{cases} x(s) = 1 - sp(s) = \mathbf{P}(S(X) > s | Y = 0) \\ y(s) = se(s) = \mathbf{P}(S(X) \geq s | Y = 1) \end{cases}$$

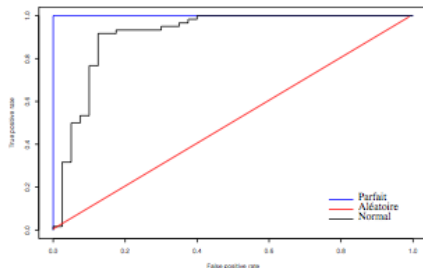


Figure :

Estimation

- La courbe ROC associée à une fonction de score S nécessite le calcul des probabilités $\mathbf{P}(S(X) > s | Y = 0)$ et $\mathbf{P}(S(X) \geq s | Y = 1)$.
- Ces probabilités ne sont pas calculables en pratique et doivent être **estimées**.
- L'estimation s'effectue à l'aide d'un échantillon **indépendant** de celui utilisé pour construire la fonction de score.

Apprentissage / Validation

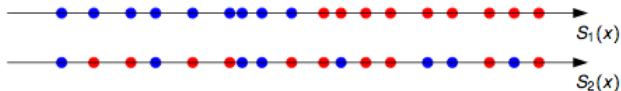
Un premier résultat

- 2 quantités sont à estimer :
 - ① La fonction de score $S(x)$
 - ② Les paramètres de la courbe ROC : $\mathbf{P}(S(X) > s | Y = 0)$ et $\mathbf{P}(S(X) \geq s | Y = 1)$.
- L'échantillon $(X_1, Y_1), \dots, (X_n, Y_n)$ est séparé deux :
 - ① **un échantillon d'apprentissage** $(X_1, Y_1), \dots, (X_\ell, Y_\ell)$ utilisé pour estimer la fonction de score (par exemple les paramètres du modèle logistique pour le score logistique).
 - ② **un échantillon test** $(X_{\ell+1}, Y_{\ell+1}), \dots, (X_n, Y_n)$ pour estimer la courbe ROC.

Apprentissage / Validation

Une fois le score S estimé à l'aide de l'échantillon d'apprentissage, les paramètres de la courbe ROC sont estimés comme suit :

- ① On calcule le score des individus de l'échantillon test.
- ② On définit un **nouvel échantillon** $(S(X_{\ell+1}), Y_1), \dots, (S(X_n), Y_n)$



- ③ Les paramètres de la courbes ROC

$$\begin{cases} x(s) = 1 - sp(s) = \mathbf{P}(S(X) > s | Y = 0) \\ y(s) = se(s) = \mathbf{P}(S(X) \geq s | Y = 1) \end{cases}$$

sont **estimés** par

$$\begin{cases} \hat{x}(s) = \frac{1}{\text{Card}\{i : Y_i = 0\}} \sum_{i: Y_i=0} \mathbf{1}_{S(X_i) > s} \\ \hat{y}(s) = \frac{1}{\text{Card}\{i : Y_i = 1\}} \sum_{i: Y_i=1} \mathbf{1}_{S(X_i) > s} \end{cases}$$

Exemple

- On reprend l'exemple sur la maladie cardiovasculaire et on **compare les 3 modèles construits** (2 logistique et un arbre) à l'aide de la **courbe ROC**.
- On calcule d'abord le **score** des individus de **l'échantillon test**.

```
> score1 <- predict(model1,newdata=dtest,type="response")  
> score2 <- predict(model2,newdata=dtest,type="response")  
> score3 <- predict(arbre,newdata=dtest)
```

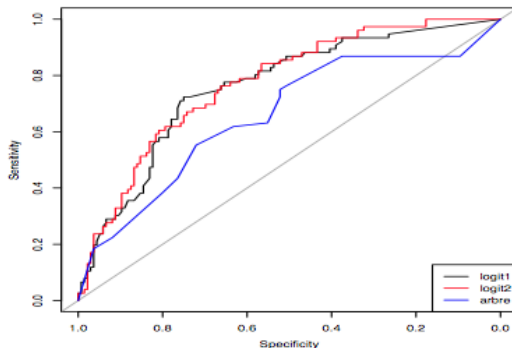
- On trace ensuite la **courbe roc** à l'aide de la fonction **roc** du package pROC.

```
> roc(dtest$chd,score1,plot=TRUE)  
> roc(dtest$chd,score2,plot=TRUE,col="red",add=TRUE)  
> roc(dtest$chd,score3,plot=TRUE,col="blue",add=TRUE)  
> legend("bottomright",legend=c("logit1","logit2","arbre"),  
        col=c("black","red","blue"),lty=1,lwd=2)
```

Figure :

Courbes ROC

Autre Exemple



Conclusion

Pour le critère ROC, les modèles logistiques sont plus performants que l'arbre de classification.

Sommaire

- 1 Sélection-Choix de modèles
- 2 Sélection de variables**
- 3 Validation de modèles
- 4 Examen des résidus

Motivations

- Dans la partie précédente, on a présenté des outils permettant de comparer des modèles **construits**.
- On se place dans un cadre différent : étant donné p variables explicatives X_1, \dots, X_p , on cherche une procédure automatique permettant de trouver le "**meilleur**" **sous-groupe de variables** à mettre dans le modèle logistique.

Pourquoi ?

(Au moins) 2 raisons peuvent motiver cette démarche :

- 1 **Descriptif** : identifier les variables qui permettent d'**expliquer la cible**.
- 2 **Statistique** : la variance des estimateurs augmente avec le nombre de paramètres du modèle. Diminuer le nombre de variables permettra d'avoir des **estimateurs plus précis**.

Figure :

Recherche exhaustive

- Une approche naturelle est de construire **tous** les modèles logistiques (2^p) et de retenir celui qui **optimise un critère donné** (AIC-BIC...).
- Les package leaps permet de faire cela pour la **régression linéaire**.
- Pour le **modèle logistique**, on peut utiliser le package bestglm.

```
> library(bestglm)
> model4 <- bestglm(dapp,family=binomial,IC="BIC")
Morgan-Tatar search since family is non-gaussian.
> model4$BestModel
```

```
Call: glm(formula = y ~ ., family = family, data = Xi, weights = weights)
```

```
Coefficients:
(Intercept)          ld1  famhistPresent          age
   -4.29645       0.18650       0.82172       0.05088
```

```
Degrees of Freedom: 249 Total (i.e. Null);  246 Residual
Null Deviance:      319.2
Residual Deviance: 267.5  AIC: 275.5
```

Exemple

- On peut également visualiser les **variables retenues dans les meilleurs modèles** pour le critère donné

```
> model4$BestModels
      sbp tobacco   ldl adiposity famhist typea obesity alcohol age Criteri
1 FALSE  FALSE  TRUE   FALSE    TRUE FALSE  FALSE  FALSE TRUE  284.04
2 FALSE  FALSE  TRUE   FALSE    FALSE FALSE  FALSE  FALSE TRUE  286.05
3 FALSE   TRUE  TRUE   FALSE    TRUE FALSE  FALSE  FALSE TRUE  286.78
4 FALSE  FALSE FALSE   FALSE    TRUE FALSE  FALSE  FALSE TRUE  287.32
5 FALSE  FALSE  TRUE   FALSE    TRUE  TRUE  FALSE  FALSE TRUE  287.91
```

Lorsque le nombre de variables p est trop grand, balayer tous les modèles peut se révéler **très coûteux en tant de calcul**. On a alors recours à des méthodes **pas à pas**.

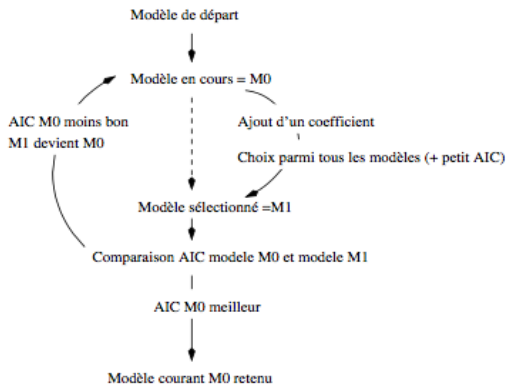
Méthodes pas à pas

L'approche consiste à :

- construire un **modèle initial**
- Ajouter (**forward**) ou supprimer (**backward**) la variable qui optimise un critère donné (**BIC** ou **AIC**) par exemple.
- Répéter le processus jusqu'à un **critère d'arrêt**.

Technique ascendante utilisant l'AIC

Très Important !



Exemple sur R

- La fonction **step** permet de sélectionner des variables à l'aide de méthodes **pas à pas**.

```
> model_complet <- glm(chd~.,data=dapp,family=binomial)
> model_step <- step(model_complet,direction="backward",k=log(nrow(dapp)))
> model_step
```

```
Call:  glm(formula = chd ~ ldl + famhist + age, family = binomial,
          data = dapp)
```

Coefficients:

(Intercept)	ldl	famhistPresent	age
-4.29645	0.18650	0.82172	0.05088

Figure :

Sommaire

- 1 Sélection-Choix de modèles
- 2 Sélection de variables
- 3 **Validation de modèles**
 - Test d'adéquation de la déviance
 - Test d'adéquation de la déviance
 - Test d'adéquation de Pearson
 - Test d'Hosmer Lameshow
- 4 Examen des résidus

Problématique

- Poser un modèle revient à faire une **hypothèse** : la loi de la variable d'intérêt appartient à une **famille de loi donnée**.
- Pour le **modèle logistique** cette hypothèse est que la loi des Y_i est une Bernoulli de paramètre $p_\beta(x_i)$ tel que

$$\text{logit } p_\beta(x_i) = \beta_1 x_{i1} + \dots + \beta_p x_{ip}.$$

- Les résultats présentés précédemment sont **vrais uniquement sous cette hypothèse**. Il faut par conséquent la vérifier.

Les techniques permettant (**dans une certaine mesure**) de vérifier cette hypothèse sont **similaires à celles du modèle de régression linéaire** (tests d'adéquation, étude des résidus...).

La déviance

- **Idée** : se baser sur la **vraisemblance**. En effet, plus la vraisemblance est proche de 1, plus le modèle est "proche" des données.
- La valeur d'une vraisemblance est difficile à interpréter (elle dépend notamment du nombre de données).
- La **déviance** permet de comparer la vraisemblance du modèle à celle d'un modèle parfait en terme d'adequation aux données : **le modèle saturé**.

Figure :



Le modèle saturé

Propriétés générales de l'emv

- C'est le modèle qui ajuste "**parfaitement**" les observations. Il faut dissocier les types de données pour le définir.

Données individuelles

On note $(x_1, Y_1), \dots, (x_n, Y_n)$ l'échantillon (tous les x_i sont différents). Le modèle saturé modélise la loi des Y_i par des Bernoulli de paramètre $p_{sat}(x_i)$ estimés selon $\hat{p}_{sat}(x_i) = Y_i$.

Données répétées

On note $(x_1, n_1, Y_1), \dots, (x_T, n_T, Y_T)$ l'échantillon. Le modèle saturé modélise la loi des Y_t par des Binomiale de paramètres $(n_t, p_{sat}(x_t))$ avec $\hat{p}_{sat}(x_t) = Y_t/n_t$.

Le modèle saturé

Théorème

- On désigne par \mathcal{L}_{sat} la log-vraisemblance du modèle saturé calculé au point défini par les $\hat{p}_{sat}(x_i)$.

Propriété

- Dans le cas de données individuelles, $\mathcal{L}_{sat} = 0$.
- Pour des données répétées, on a

$$\mathcal{L}_{sat} = \sum_{t=1}^T \log \binom{n_t}{y_t} + \sum_{t=1}^T y_t \log \hat{p}_{sat}(x_t) + (n_t - y_t) \log(1 - \hat{p}_{sat}(x_t)).$$

Remarque

- En terme d'ajustement, on ne peut pas faire mieux que le modèle saturé.
- Néanmoins, ce modèle n'est généralement pas bon : il est **sur-paramétré** (il contient autant de paramètres que de points d'où son nom).

La Déviance

Théorème

- On note \mathcal{M} un modèle logistique, $\hat{\beta}_n$ l'emv des paramètres et \mathcal{L}_n la log-vraisemblance de ce modèle.

Définition

La déviance de \mathcal{M} est définie par

$$D_{\mathcal{M}} = 2(\mathcal{L}_{sat} - \mathcal{L}_n(\hat{\beta}_n)).$$

- La déviance est positive $D_{\mathcal{M}} \geq 0$.
- Plus la déviance est faible, meilleur est le modèle en terme d'ajustement.

Illustration

- On reprend le jeu de données sur le "role des femmes" dans la société (données répétées).
- La déviance est présente dans les sorties de la fonction **glm** :

```
> modell <- glm(cbind(agree, disagree)~sex+education, data=womensrole, famil
> modell
```

```
Call: glm(formula = cbind(agree, disagree) ~ sex + education,
      family = binomial, data = womensrole)
```

Coefficients:

(Intercept)	sexFemale	education
2.74796	-0.04349	-0.28970

Degrees of Freedom: 29 Total (i.e. Null); 27 Residual

Null Deviance: 398.9

Residual Deviance: 36.89 AIC: 165.4

- On peut également la récupérer avec la fonction **deviance**

```
> deviance(modell)
[1] 36.89419
```

Le test d'adéquation de la déviance

- **Idée** : déviance faible \implies bonne adéquation.
- On pose H_0 : "le modèle est adéquat" (les données sont bien générées selon le modèle logistique en question) contre H_1 : "il ne l'est pas".

Propriété

En présence de **données répétées**, la déviance suit une loi du χ^2_{T-p} sous H_0 lorsque $n_t \rightarrow \infty, t = 1, \dots, T$.

- **Conclusion** : on rejette H_0 si $D_{M,obs}$ est plus grande que le quantile d'ordre $1 - \alpha$ de la loi χ^2_{T-p} .
- Sur R, on calcule la **probabilité critique** avec

```
> 1-pchisq(modell$deviance,modell$df.resid)
[1] 0.09705949
```

Le test d'adéquation de Pearson

- Permet de tester les mêmes hypothèses que précédemment et toujours pour des **données répétées**.
- La **statistique de test** est la suivante :

$$P = \sum_{t=1}^T \frac{(y_t - n_t p_{\hat{\beta}_n}(x_t))^2}{n_t p_{\hat{\beta}_n}(x_t) (1 - p_{\hat{\beta}_n}(x_t))}$$

Propriété

En présence de **données répétées**, P suit une loi du χ^2_{T-p} sous H_0 en présence de données répétées lorsque $n_t \rightarrow \infty, t = 1, \dots, T$.

- **Conclusion** : on rejette H_0 si P_{obs} est plus grande que le quantile d'ordre $1 - \alpha$ de la loi χ^2_{T-p} .
- Sur R, on calcule la **probabilité critique** avec


```
> P <- sum(residuals(modell,type="pearson")^2)
> 1-pchisq(P,nrow(womensrole)-length(modell$coef))
```


Commentaires

- Les deux tests d'adéquation sont **asymptotiques** et utilisables **uniquement dans le cas de données répétées**.
- Il faut par conséquent avoir **suffisamment d'observations en chaque points du design** pour pouvoir les appliquer.
- Le test de déviance est généralement privilégié.
- En présence de données individuelles, on utilise souvent le **test de Hosmer et Lemeshow** : l'approche consiste à **regrouper les données** et à définir une statistique de test de type Pearson.

Figure :

Test d'Hosmer Lameshow

On est en présence de données individuelles $(x_1, Y_1), \dots, (x_n, Y_n)$. La statistique de test se construit comme suit.

- ① Les probabilités estimées $p_{\hat{\beta}_n}(x_i)$ sont **ordonnées par ordre croissant**.
- ② Ces probabilités ordonnées sont ensuite **séparées en K groupes** de taille égale (on prend souvent $K = 10$ si n est suffisamment grand). On note
 - m_k les effectifs du groupe k ;
 - o_k le nombre de succès ($Y = 1$) observé dans le groupe k ;
 - μ_k la moyenne des $\hat{p}_\beta(x_i)$ dans le groupe k .

- La statistique de test est alors

$$C^2 = \sum_{k=1}^K \frac{(o_k - m_k \mu_k)^2}{m_k \mu_k (1 - \mu_k)}.$$

- Le test se conduit de manière identique au test de déviance, la statistique C^2 suivant approximativement sous H_0 un χ^2_{K-2} .

Sommaire

- 1 Sélection-Choix de modèles
- 2 Sélection de variables
- 3 Validation de modèles
- 4 Examen des résidus

Pourquoi ?

- L'analyse des résidus permet, dans une certaine mesure, d'affiner un modèle.
- Elle permet de détecter des individus atypiques ou aberrants ou encore de détecter des effets non linéaires.
- On distingue plusieurs types de résidus que nous présentons dans le cas de données répétées.

Les résidus de Pearson

On désigne par $(x_t, n_t, Y_t), t = 1, \dots, T$ les données.

- Les résidus de Pearson sont définis par :

$$Rp_t = \frac{Y_t - n_t p_{\hat{\beta}_n}(x_t)}{\sqrt{n_t p_{\hat{\beta}_n}(x_t)(1 - p_{\hat{\beta}_n}(x_t))}}.$$

- Lorsque n_t est grand, la loi de Rp_t est proche d'une $\mathcal{N}(0, 1)$. On peut ainsi analyser les résidus de Pearson de la même manière que les résidus du modèle linéaire Gaussien.
- La statistique de Pearson s'exprime en fonction des résidus de Pearson $P = \sum_{t=1}^T Rp_t^2$.

Version Standardisée

- Les résidus de Pearson définis précédemment ne sont **pas de variance 1**.
- Il est souvent préférable d'utiliser une **version standardisée** de ces résidus. Pour ce faire, on remarque que

$$V[Y_t - np_{\hat{\beta}_n}(x_t)] \approx n_t p_{\hat{\beta}_n}(x_t)(1 - p_{\hat{\beta}_n}(x_t))(1 - h_t),$$

où h_t est le terme élément de la diagonale de

$$H = X(X'W_{\hat{\beta}_n}X)^{-1}X'W_{\hat{\beta}_n}.$$

Définition

Les **résidus de Pearson standardisés** sont définis par

$$Rps_t = \frac{Y_t - n_t p_{\hat{\beta}_n}(x_t)}{\sqrt{n_t p_{\hat{\beta}_n}(x_t)(1 - p_{\hat{\beta}_n}(x_t))(1 - h_t)}}, t = 1, \dots, T.$$

Résidus de déviance

- Les **résidus de déviance** sont définis par

$$Rd_t = \sqrt{2 \left[y_t \log \frac{\bar{Y}_t}{p_{\hat{\beta}_n}(x_t)} + (n_t - Y_t) \log \frac{n_t - Y_t}{n_t - n_t p_{\hat{\beta}_n}(x_t)} \right]}.$$

- Lorsque n_t est grand, la loi de Rd_t est proche d'une $\mathcal{N}(0, 1)$. On peut ainsi analyser les résidus de déviance de la même manière que les résidus du modèle linéaire Gaussien.
- La **déviance** s'exprime en fonction des résidus de déviance $D = \sum_{t=1}^T Rd_t^2$.

- Là encore, il existe une version standardisée :

$$Rds_t = \frac{Rd_t}{\sqrt{1 - h_t}}, \quad t = 1, \dots, T.$$

Figure :

Examen des résidus

- Les diagnostics sont essentiellement **graphiques** :
 - ① **Index plot** : numéro de l'observation en abscisse, valeur du résidu en ordonnée.
 - ② **Prédiction/résidus** : probabilité prédite au point x_t en abscisse et résidu en ordonnée.
- On pourra identifier :
 - ① Les valeurs **élevées de résidus** (individus atypiques...)
 - ② **Structures sur le nuage des résidus** (si c'est le cas il faudra envisager de modifier la combinaison linéaire des variables explicatives)

Remarque importante

- Les résultats énoncés sur les résidus (Pearson ou déviance) sont vraies lorsque n_t est grand...
- Dans le cas de données individuelles, on observera (quasi)-systématiquement des structurations sur les nuages de résidus.

Figure :

Tracé des résidus

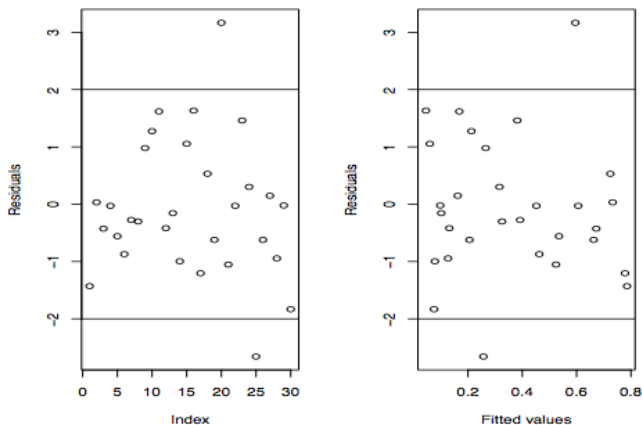


Figure :

Résidus partiels

- On considère le modèle logistique

$$\text{logit } p_{\beta}(x) = \beta_1 x_1 + \dots, \beta_p x_p.$$

- Les résidus partiels sont définis par :

$$r_{tj} = \frac{Y_t - n_t p_{\hat{\beta}_n}(x_t)}{n_t p_{\hat{\beta}_n}(x_t)(1 - p_{\hat{\beta}_n}(x_t))} + \hat{\beta}_j x_{tj}, \quad t = 1, \dots, T, j = 1 \dots p.$$

Diagnostic

- L'analyse consiste à tracer pour **toutes les variables j** les T résidus r_{tj} , $t = 1, \dots, T$.
- Si le tracé est linéaire alors tout est "normal". Si par contre une **tendance non linéaire se dégage**, il faut remplacer la variable j par une fonction de celle ci donnant la même tendance que celle observée.

Résidus partiels

- On considère le modèle logistique permettant d'expliquer `etat` par `marque` et `age` pour les données `panne`.
- La fonction **residuals** permet de calculer les **résidus partiels**

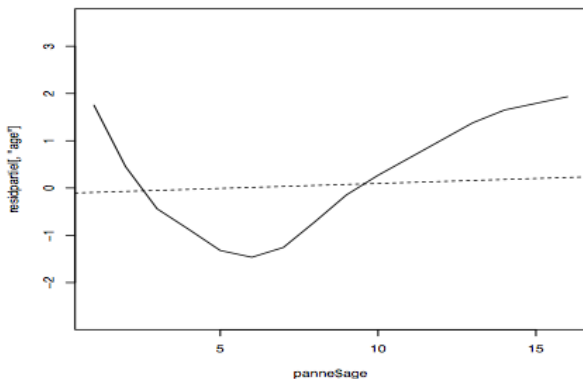
```
> model <- glm(etat~.,data=panne,family=binomial)
> residpartiel <- residuals(model,type="partial")
```

- On trace les résidus partiels pour la variable `age` avec :

```
> plot (panne$age,residpartiel[, "age"],cex=0.5)
> est <- loess(residpartiel[, "age"]~panne$age)
> ordre <- order(panne$age)
> matlines(panne$age[ordre],predict(est)[ordre])
> abline(lsfite(panne$age,residpartiel[, "age"]),lty=2)
```

Figure :

Tracé des résidus partiels



Conclusion

Le graphe suggère d'ajouter la variable age^2 dans le modèle.

Points leviers

- Ce sont les points du design qui déterminent **fortement leur propre estimation**.
- L'analyse est **similaire à celle du modèle de régression linéaire**.

- On rappelle que l'emv $\hat{\beta}_n$ s'écrit

$$\hat{\beta}_n = (\mathbf{X}' \mathbf{W}_{\hat{\beta}} \mathbf{X})^{-1} \mathbf{X}' \mathbf{W}_{\hat{\beta}} \mathbf{Z}.$$

- La prédiction linéaire des individus est donc donnée par

$$\mathbf{X} \hat{\beta}_n = \mathbf{X} (\mathbf{X}' \mathbf{W}_{\hat{\beta}} \mathbf{X})^{-1} \mathbf{X}' \mathbf{W}_{\hat{\beta}} \mathbf{Z} = \mathbf{H} \mathbf{Z},$$

- Et celle de l'individu i par

$$[\mathbf{X} \hat{\beta}_n]_i = H_{ii} Z_i + \sum_{j \neq i} H_{ij} Z_j.$$

Points leviers

- H étant un projecteur, on a $0 \leq H_{ii} \leq 1$. Par conséquent
 - Si $H_{ii} = 1$, alors $\hat{p}_{\hat{\beta}_n}(x_i)$ est entièrement déterminé par la i ème observation.
 - Si $H_{ii} = 0$, la i ème observation n'influence pas $\hat{p}_{\hat{\beta}_n}(x_i)$.

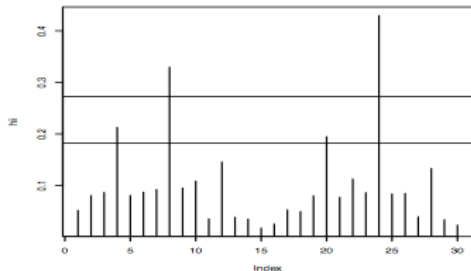
Conclusion

- Pour mesurer l'influence d'une observation sur sa propre estimation, on représente le diagramme en batons des H_{ii} .
- On compare généralement la valeur des H_{ii} à $2p/n$ ou $3p/n$ pour déclarer les points comme **leviers**.

Exemple

- On trace le diagramme en bâton des H_{ii} pour le modèle construit sur les données **womensrole**.

```
> model <- glm(cbind(agree,disagree)~sex+education,data=womensrole,  
               family=binomial)  
  
> p <- length(model$coef)  
> n <- nrow(panne)  
> plot(influence(model)$hat,type="h",ylab="hii")  
> abline(h=c(2*p/n,3*p/n))
```



Poins Influent

- Les **points influents** sont des points qui influent sur le modèle de telle sorte que si on les enlève, alors l'**estimation des coefficients sera fortement changée**.
- La mesure la plus classique d'influence est la **distance de Cook**. Il s'agit d'une distance entre le coefficient estimé avec **toutes les observations** et celui estimé avec toutes les observations sauf une.

Définition

La distance de Cook pour l'individu i est définie par

$$DC_i = \frac{1}{p} (\hat{\beta}_{(i)} - \hat{\beta}_n)' \mathbf{X}' \mathbf{W}_{\hat{\beta}} \mathbf{X} (\hat{\beta}_{(i)} - \hat{\beta}_n) \approx \frac{r_{Pi}^2 H_{ii}}{p(1 - H_{ii})^2},$$

où r_{Pi} est le résidu de Pearson pour le i ème individu et $\hat{\beta}_{(i)}$ est l'emv calculé sans la i ème observation.

Distance de Cook

- Là encore, on représente la distance de Cook de chaque point du design à l'aide d'un **diagramme en batons**.
- Si une distance se révèle **grande par rapport aux autres**, alors ce point sera considéré comme **influent**. Il convient alors de comprendre pourquoi il est influent :
 - il est levier ;
 - il est aberrant ;
 - (les deux !)

Dans tous les cas il convient de **comprendre si une erreur de mesure, une différence dans la population des individus est à l'origine de ce phénomène**. Eventuellement pour obtenir des conclusions robustes il sera bon de **refaire l'analyse sans ce(s) point(s)**.

Exemple

- La fonction **cooks.distance** permet de calculer les distances de Cook sur R :

```
> plot(cooks.distance(model), type="h", ylab="Distance de Cook")
```

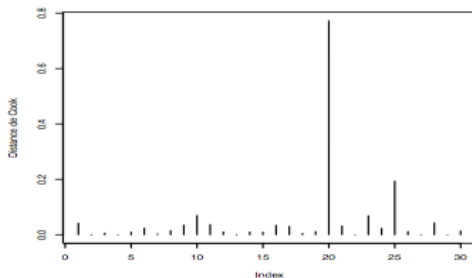


Figure :