Scénario : sélection de modèle par sélection de variables et pénalisation

Résumé

Comparaison sur le même jeu de données (comptabilité d'entreprises) des qualités de prévision de plusieurs modèles linéaires obtenus par :

– *Sélection de variables (critères CP, BIC)*
– *Régularisation (ridge)*
– *Régression sur composantes (PCR, PLS)*

Les mêmes calculs sont réalisés successivement avec SAS puis avec R.

1 Introduction

1.1 Objectif

Les grands principes et les questions fondamentales des problèmes de choix de modèles sont abordés dans le cadre plus pédagogique du modèle gaussien ou de régression classique. Aussi, dans le cas de données présentant un problème de multicolinéarité, il s'agit de comparer les techniques visant à la recherche de modèles de régression linéaire multiple *parcimonieux* avec celle de régression biaisée (*ridge* ou encore de régression sur des facteurs (régression sur composantes principales, PLS).

Les qualités prédictives de tous les modèles obtenus seront comparés sur un échantillon test. Toujours par souci pédagogique, l'analyse est conduite d'abord avec SAS puis avec R.

1.2 Les données

Les données (Jobson, 1991) décrivent les résultats comptables de 80 entreprises du Royaume Uni. RETCAP est la variable à modéliser. Les entreprises sont réparties aléatoirement en deux groupes de 40 entreprises.

Descriptif des 13 variables :

RETCAP	Return on capital employed
WCFTDT	Ratio of working capital flow to total debt
LOGSALE	Log to base 10 of total sales
LOGASST	Log to base 10 of total assets
CURRAT	Current ratio
QUIKRAT	Quick ratio
NFATAST	Ratio of net fixed assets to total assets
FATTOT	Gross sixed assets to total assets
PAYOUT	Payout ratio
WCFTCL	Ratio of working capital flow to total current liabilities
GEARRAT	Gearing ratio (debt-equity ratio)
CAPINT	Capital intensity (ratio of total sales to total assets)
INVTAST	Ratio of total inventories to total assets

2 Prise en charge avec SAS

2.1 Lecture des données

Charger les données du fichier `ukcomp1.dat` à partir de l'URL :

`http://wikistat/data/`

Exécuter le programme de lecture :

```
data sasuser.ukcomp1 ;
infile 'ukcomp1.dat' dlm='09'x;
input RETCAP GEARRAT CAPINT WCFTDT LOGSALE LOGASST
 CURRAT QUIKRAT NFATAST INVTAST FATTOT PAYOUT WCFTCL;
poids=1;
run;
```

2.2 Exploration

Vérifier rapidement les données dans le module SAS/INSIGHT (menu `solutions>analysis>interactive data analysis`, sélection de la table), l'allure raisonnablement symétrique des distributions, la présence de quelques points atypiques.

3 SAS Choix de modèle par sélection de variables

3.1 Modèle complet

Estimer dans SAS/INSIGHT (`Analyse>fit` le modèle complet sur le premier fichier expliquant RETCAP (sélectionnée dans Y) avec toutes les autres variables (sélectionnées dans X).

Retrouver les résultats fournis par la procédure classique SAS/REG utilisée dans le programme suivant. Beaucoup d'options y sont actives afin de fournir la plupart des résultats même si certains sont redondants ou peu utiles. Faire attention aux résidus studentisés.

```
proc reg data=sasuser.ukcomp1 all;
 model RETCAP=WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
   NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT
   /dw covb Influence cli clm tol vif collin R P;
output out=resout h=lev p=pred r=res student=resstu ;
run;
```

3.2 Choix de modèle "à la main" par élimination

SAS propose des algorithmes de sélection automatique des variables. Néanmoins il est nécessaire de savoir se "débrouiller" avec les outils plus limités proposés par d'autres logiciels.

Itérer la procédure suivante dans SAS/INSIGHT :

1. Choisir, parmi les variables explicatives, celle X^j pour lequel le test de Student ($H_0 : b_j = 0$) est le moins significatif, c'est-à-dire avec la plus grande "prob value".
2. La retirer du modèle et recalculer l'estimation. Il suffit pour cela de sélectionner le nom de la variable dans le tableau (TYPE III) et d'exécuter la commande `delete` du menu `edit` de la même fenêtre. Le modèle est ré-estimé automatiquement.

Arrêter le processus lorsque tous les coefficients sont considérés comme significativement (à 5%) différents de 0. Attention, la "variable" INTERCEPT (terme constant) ne peut pas être considérée au même titre que les autres variables ; la *conserver* toujours dans le modèle.

3.3 Procédures automatiques

Noter la séquence des modèles ainsi obtenus. Comparer avec la procédure automatique identique descendante :

```
proc reg data=sasuser.ukcomp1 ;
model RETCAP = WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
   NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT
   / selection=backward; /* choix de la procédure */
run;
```

Comparer avec la procédure automatique ascendante par ajout de variables :

```
proc reg data=sasuser.ukcomp1 ;
model RETCAP = WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
   NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT
   / selection=forward; /* choix de la procédure */
run;
```

3.4 Optimisation globale

Parmi les trois types d'algorithmes disponibles dans SAS et les différents critères de choix, une des façons les plus efficaces consistent à choisir les options du programme ci-dessous. Tous les modèles (parmi les plus intéressants selon l'algorithme de Furnival et Wilson) sont considérés. Seul le meilleur pour chaque niveau, c'est-à-dire pour chaque valeur p du nombre de variables explicatives sont donnés. Il est alors facile de choisir celui minimisant l'un des critères globaux (C_p, BIC...) estimant un risque pénalisé.

```
proc reg data=sasuser.ukcomp1 ;
model RETCAP = WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
   NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT
   / selection=rsquare cp adjrsq bic best=1;
run;
```

Sélectionner le modèle de C_p minimum et celui de R^2 ajusté maximum. Une autre procédure peut encore être testée. Elle correspond à l'option : `selection=stepwise`.

3.5 Dernières estimations

Estimer les différents modèles : choix descendant, meilleur C_p, meilleur R^2 ajusté. On pourrait aussi retenir celui minimisant PRESS c'est-à-dire mi-

nimisant l'estimation par validation croisée de l'erreur de prédiction, mais la procédure n'est pas automatique dans SAS.

```
proc reg data=sasuser.ukcomp1 all;
model RETCAP = ... /* liste de variables des trois
     modèles précédemment retenus /* / collin r p ;
run;
```

Vérifier sur ces derniers modèles les valeurs du diagnostic global de colinéarité.

Attention, la validité du modèle ainsi obtenu reste conditionnée à celle de l'hypothèse de *linéarité*. Il peut posséder d'honnêtes propriétés *prédictives* sans pour autant avoir des capacités d'*explication* de la variable RETCAP. De même, les estimateurs inférentiels (intervalles de confiance) sont dépendants de l'hypothèse d'homoscédasticité.

4 SAS Comparaison des modèles sur un échantillon test

L'objet de cette section est de comparer plusieurs méthodes de sélection de modèles dans le cas d'un problème de multicolinéarité : par sélection de variables (cf. ci-dessus), par régression biaisée (ridge), par régression sur composantes principales, par régression PLS (partial least square).

Nous disposons de deux échantillons. Le premier, dit échantillon d'apprentissage, sert à rechercher un meilleur modèle pour chacune des méthodes et à estimer les paramètres de ce modèle. Chacun de ces modèles sont ensuite appliqués au deuxième échantillon, dit échantillon test, pour prédire les valeurs de la variable à expliquée. Une estimation de l'erreur de prévision : la somme des carrés des différences entre valeurs prédites et valeurs observées, renseigne sur la qualité d'un modèle et permet de les comparer entre eux et donc de comparer les différentes méthodes de régression.

4.1 Lecture des données

Charger les données de l'échantillon test à partir du fichier `ukcomp2.dat`.

Lire les données de l'échantillon test ; attention à l'ordre des variables, il n'est pas le même.

```
data sasuser.ukcomp2 ;
infile 'ukcomp2.dat' dlm='09'x;
input RETCAP2 WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
  NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT;
poids=0;
run;
```

Concaténer verticalement les fichiers

```
data sasuser.ukcomp;
set sasuser.ukcomp1 sasuser.ukcomp2;
run;
```

4.2 Modèles obtenus par sélection de variables

Pour estimer le modèle sur les 40 premières observations et prévoir les 40 suivantes, il suffit d'estimer dans INSIGHT (ou avec la procédure REG) le modèle expliquant RETCAP. Les calculs sont faits en excluant les données manquantes et donc sur les 40 premières observations car les 40 suivantes ont des valeurs manquantes pour la variable RETCAP. En revanche, les prédictions sont calculées pour toutes donc sur les 40 dernières ou échantillon test.

Comparer les valeurs prédites et les valeurs observées de RECAP2. Pour cela, calculer la somme des carrés des erreurs pour plusieurs modèles : le modèle complet, celui qui maximise le R^2 ajusté, celui qui minimise le C_p.

La somme des carrés des erreurs est simplement calculée dans INSIGHT en étudiant la distribution de la nouvelle variable P_RECAP−RECAP2 des écarts entre observations sur les 40 entreprises de test et valeurs prédites. Le paramètre `USS` (unmodified sum of squares) fournit la bonne valeur.

4.3 Régression ridge

La sélection de variables permet donc de restreindre les problèmes de colinéarité, source importante de variance des prédictions. Une autre façon de résoudre ce problème consiste à calculer une estimation sous contrainte sur la norme du vecteur des paramètres ou, c'est équivalent, à "translater" d'une valeur k la diagonale de la matrice à inverser afin d'améliorer son conditionnement. Cette technique dire de *régularisation* est encore appelée *ridge regression* ; elle est calculée par la procédure `reg` ci-dessous.

```
proc reg data=sasuser.ukcomp1 ridge= 0 to 0.2 by 0.01
```

```
   outest=ridgest;
model RETCAP=WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
   NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT
         / noprint;
plot / ridgeplot nomodel nostat vref=0 lvref=1
   cvref=blue cframe=ligr;
Proc print; run;
quit;
```

Les paramètres estimés pour chaque valeur de k sont dans la table *work.ridgest*. Problème : Comment rechercher la valeur "optimale" de k car SAS/STAT ne prévoit pas de procédure automatique de type validation croisée. On se sert du graphe représentant les valeurs des paramètres en fonction du coefficient de *ridge* ; il est obtenu par l'option *ridgeplot*. La valeur 0.03 semble raisonnable en première approximation.

Après avoir estimé le modèle pour la valeur optimale, la procédure `score` est utilisée pour calculer les prévisions du modèle choisi.

```
proc reg data=sasuser.ukcomp1 outest=ridgest;
model RETCAP = WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
   NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT /
         ridge= 0.03 noprint;
proc score data=sasuser.ukcomp score=ridgest
  out=ridgeprev predict type=ridge;
var WCFTCL WCFTDT GEARRAT LOGSALE LOGASST NFATAST CAPINT
    FATTOT INVTAST PAYOUT QUIKRAT CURRAT;
run;
```

Comme dans le cas précédent, calculer la somme des carrés des erreurs de prévision. Ouvrir dans SAS/Insight la table work.ridgeprev puis calculer la variable `retcap2-model1` puis la somme des carrés.

4.4 Régression sur composantes principales

L'approche suivante qui peut, dans certaines situations, donner de bons résultats se déroule en deux étapes.

1. Calcul des "variables principales" deux à deux orthogonales et engendrant le même espace que les variables explicatives par une analyse en composantes principales,
2. Régression sur ces variables principales après une sélection automatique des variables.

Attention à une complication : l'ACP est calculée sur l'ensemble des deux échantillons avec, pour les observations de l'échantillon test, un poids nul. Ainsi, ces observations ne participent pas aux calculs des axes mais leurs coordonnées (composantes principales) sont évaluées comme points supplémentaires dans la base des vecteurs propres.

```
proc princomp data=sasuser.ukcomp out=comp;
var WCFTCL WCFTDT GEARRAT LOGSALE LOGASST NFATAST CAPINT
    FATTOT INVTAST PAYOUT QUIKRAT CURRAT;
weight poids;
proc reg data=comp;
model retcap=prin1--prin12/selection=rsquare cp best=1;
run;quit;
```

Comparer les valeurs des C_p et déterminer les variables ou composantes principales à retenir. Noter que les variables principales les plus importantes pour la prédiction ne sont pas nécessairement celles de plus grande variance. Calculer les erreurs de prévision sur le deuxième échantillon d'entreprises en ré-estimant le modèle sélectionné dans SAS/INSIGHT à partir des données de la table des composantes principales : `work.comp`.

4.5 Régression PLS

Cette dernière approche permet d'illustrer l'usage de la régression PLS très utilisée dans des situations de multicolinéarité et même lorsque le nombre de variables explicatives excède le nombre d'observations. C'est le cas par exemple en chimiométrie lorsque les variables explicatives sont issues de la discrétisation de mesures spectrales (inra-rouge ou HPLC).

Une première exécution demande un nombre important de facteurs ainsi qu'une estimation par validation croisée de l'erreur de prédiction.

```
proc pls data=sasuser.ukcomp1 cv=one nfac=10;
model retcap=WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
  NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT;
run;
```

Le nombre de facteurs est sélectionné comme étant celui qui minimise l'estimation de l'erreur par validation croisée (voir la sortie). Le modèle est ensuite

ré-estimé en fixant le nombre de facteurs et en introduisant le 2ème groupe d'observations pour calculer l'erreur de prédiction sur ce groupe.

```
proc pls data=sasuser.ukcomp nfac=2;
model retcap=WCFTCL WCFTDT GEARRAT LOGSALE LOGASST
  NFATAST CAPINT FATTOT INVTAST PAYOUT QUIKRAT CURRAT;
output out=plstat predicted=ychap;
run;
```

Comme précédemment, ouvrir la table work.plstat dans sas/insight pour calculer la somme des carrés des erreurs de prédiction sur le deuxième ensemble d'observations.

4.6 Conclusion

Comparer les erreurs de prédictions de chaque modèle sur l'échantillon test. Quelle est la meilleure stratégie sur ces données ?

Attention, en fonction de l'exemple traité et du jeu de données, une méthode peut apparaître meilleure qu'une autre sans généralisation possible.

5 Prise en charge avec R

Charger et lire les fichiers.

```
ukcomp.app=read.table("ukcomp1_r.dat",header=T)
ukcomp.test=read.table("ukcomp2_r.dat",header=T)
summary(ukcomp.test)
summary(ukcomp.app)
# Attention, à l'ordre des variables
ukcomp.test=data.frame(ukcomp.test
   [,names(ukcomp.app)])
```

Une ACP pour analyser la structure de corrélation des variables.

```
cor(ukcomp.app[,-1])
library(FactoMineR)
PCA(ukcomp.app)
```

Modèle linéaire complet

Une fonction utile de graphe des résidus.

```
plot.res=function(x,y,titre="")
{
plot(x,y,col="blue",ylab="Résidus",
   xlab="Valeurs predites",main=titre)
abline(h=0,col="green")
}
```

Estimation du modèle et graphes des résidus.

```
fit.lm=lm(RETCAP~.,data=ukcomp.app)
summary(fit.lm)
#Regroupement des graphiques sur la meme page
par(mfrow=c(2,2))
#Residus et points influents
plot(fit.lm,las=1)
summary(fit.lm) # noter les p-valeurs
par(mfrow=c(1,1)) # retour au graphique standard
```

6 R Sélection de modèle par sélection de variables

Sélection par AIC et backward

```
uk.lmback=step(fit.lm) # noter q
# des paramètres restent non significatifs
anova(uk.lmback)
```

Sélection par BIC et backward

```
#  k=log(n) pour BIC au lieu de AIC.
uk.lmback=step(fit.lm,k=log(40))
anova(uk.lmback) # noter q et les variables
```

Sélection par AIC et forkward

```
fit.lm=lm(RETCAP  ~ 1 , data = ukcomp.app)
uk.lmfor=step(fit.lm,scope=list(lower=~1,
  upper=~WCFTCL+WCFTDT+GEARRAT+LOGSALE+
  LOGASST+NFATAST+CAPINT+FATTOT+INVTAST+
  PAYOUT+QUIKRAT+CURRAT),direction="forward")
anova(uk.lmfor) # noter q
```

Sélection par BIC et forkward

```
fit.lm=lm(RETCAP  ~ 1 , data = ukcomp.app)
uk.lmfor=step(fit.lm,scope=list(lower=~1,
  upper=~WCFTCL+WCFTDT+GEARRAT+LOGSALE+
  LOGASST+NFATAST+CAPINT+FATTOT+INVTAST+
  PAYOUT+QUIKRAT+CURRAT),direction="forward",
  k=log(40))
anova(uk.lmfor) # noter q et les variables
```

Sélection par AIC et stepwise

```
fit.lm=lm(RETCAP  ~ 1 , data = ukcomp.app)
uk.lmboth=step(fit.lm,scope=list(lower=~1,
  upper=~WCFTCL+WCFTDT+GEARRAT+LOGSALE+
  LOGASST+NFATAST+CAPINT+FATTOT+INVTAST+
  PAYOUT+QUIKRAT+CURRAT),direction="both")
anova(uk.lmboth) # noter q, les variables
```

Sélection par BIC et stepwise

```
fit.lm=lm(RETCAP  ~ 1 , data = ukcomp.app)
uk.lmboth=step(fit.lm,scope=list(lower=~1,
  upper=~WCFTCL+WCFTDT+GEARRAT+LOGSALE+
  LOGASST+NFATAST+CAPINT+FATTOT+INVTAST+
  PAYOUT+QUIKRAT+CURRAT),direction="both",
  k=log(40))
anova(uk.lmboth)  # noter q, les variables
```

Commentaires sur le mode de sélection des algorithmes et l'importance de la pénalisation appliquée par chaque critère. Tentative de les départager par l'algorithme de Furnival et Wilson qui explore potentiellement toutes les possibilités. L'algorithme est associé au C_p de Mallows.

Recherche exhaustive et Cp

```
library(leaps)
par(mfrow=c(1,1))
#Extraction des variables explicatives
ukcomp=ukcomp.app[,2:13]
#Recherche du meilleur modèle pour chaque q
uk.choix=leaps(ukcomp,ukcomp.app[,"RETCAP"],
    method="Cp",nbest=1)
uk.choix$Cp #valeurs des Cp du meilleur modèle
plot(uk.choix$size-1,uk.choix$Cp)
# Fixer la dimension  / complexité optimale
t=(uk.choix$Cp==min(uk.choix$Cp))
# Liste des variables explicatives
colnames(ukcomp)[uk.choix$whi[t]]
```

D'autres stratégies (R^2 ajusté) conduiraient encore à d'autres modèles. Retenir celui ci-dessous minimisant le Cp.

Recherche exhaustive et Cp

```
lm.uk0=lm(RETCAP ~ WCFTDT+LOGSALE+NFATAST+CURRAT,
   data=ukcomp.app)
mean((predict(lm.uk0,newdata=ukcomp.test)-ukcomp.test
   [,"RETCAP"])**2)
```

7 R Sélection de modèle et projection sur composantes orthogonales

7.1 Régression PLS

```
library(pls)
# nombre optimal de composantes par
# validation croisée
uk.simpls= mvr(RETCAP~.,data=ukcomp.app, ncomp=12,
```

```
   validation="CV", method="simpls")
summary(uk.simpls)
#graphique
plot(uk.simpls)
#noter le nombre optimal de composantes
#Calcul des prévisions
pred.uk=predict(uk.simpls,as.matrix
(ukcomp.test[,2:13]),4)
mean((pred.uk-ukcomp.test[,"RETCAP"])**2)
```

7.2 Régression sur composantes principales

```
uk.pcr = pcr(RETCAP~.,data=ukcomp.app, ncomp=12,
   validation="CV")
summary(uk.pcr) # noter le nombre optimal
#Calcul des prévisions
pred.uk=predict(uk.pcr,as.matrix
   (ukcomp.test[,2:13]),8)
mean((pred.uk-ukcomp.test[,"RETCAP"])**2)
```

Il peut arriver que la régression sur composantes principales ne soit pas adaptée, si les premières composantes principales trouvées n'ont que peu de rapport avec la variable Y.

8 R Sélection de modèle par pénalisation

8.1 Pénalisation ridge

Comportement des coefficients

Calcul des coefficients pour différentes valeurs du paramètre lambda.

```
library(MASS)
ridge.uk=lm.ridge(RETCAP ~  .,data=ukcomp.app,
    lambda=seq(0,0.4,0.001))
par(mfrow=c(1,1))
plot(ridge.uk)
```

Pénalisation optimale par GCV

```
select(ridge.uk) # noter la valeur puis estimer
ridgeopt.uk=lm.ridge(RETCAP ~  .,data=ukcomp.app,
  lambda=0.033)
```

On peut aussi utiliser une fonction explicite de validation croisée pour tracer l'erreur en fonction de lambda.

Prévision et erreur

Pour des raisons obscures, la fonction `predict.ridgelm` n'existe pas, il faut calculer les valeurs ajustées et les prévisions à partir des coefficients.

```
coeff=coef(ridgeopt.uk)
fit.rid=rep(coeff[1],nrow(ukcomp.app))+
    as.vector(coeff[-1]%*%
    t(data.matrix(ukcomp.app[,-1])))
plot(fit.rid,ukcomp.app[,"RETCAP"])
res.rid=fit.rid-ukcomp.app[,"RETCAP"]
plot.res(fit.rid,res.rid,titre="")
```

Prévision de l'échantillon test

```
prediction=rep(coeff[1],nrow(ukcomp.test))+
   as.vector(coeff[-1]%*%t(data.matrix
   (ukcomp.test[,-1])))
mean((ukcomp.test[,1]-prediction)^2)
```

8.2 Pénalisation Lasso

Les résultats sont obtenus par la librairie `lasso2`.

Construction du modèle

```
library(lasso2)
lasso.uk=l1ce(RETCAP~.,data=ukcomp.app,bound=(1:30)/30,
trace=TRUE,absolute.t=FALSE)
```

La borne est ici relative, elle correspond à une certaine proportion de la norme $\mathbb{L}_1$ du vecteur des coefficients des moindres carrés. Une borne égale à 1 correspond donc à l'absence de pénalité, on retrouve l'estimateur des moindres carrés.

Visualisation des coefficients

```
plot(lasso.uk)
```

Sélection de la pénalité par validation croisée

```
gg.uk=gcv(lasso.uk)
gcv.uk=gg.uk[,4]
min(gcv.uk)
lasso.uk.select=l1ce(RETCAP ~ .,data=ukcomp.app,bound=27/30,
   absolute.t=FALSE)
coef=coef(lasso.uk.select)
```

Prévision et erreur

```
fit.lasso= coef[1]+   as.vector(coef[-1]%*%
    t(data.matrix(ukcomp.app[,-1])))

plot(fit.lasso,ukcomp.app[,"RETCAP"])
abline(0,1)

res.lasso=fit.lasso-ukcomp.app[,"RETCAP"]
plot.res(fit.lasso,res.lasso,titre="Residus Lasso")
```

Prévision de l'échantillon test

```
pred.lasso= coef[1]+   as.vector(coef[-1]%*%
    t(data.matrix(ukcomp.test[,-1])))

mean((pred.lasso-ukcomp.test[,"RETCAP"])^2)
```