Transfer learning avec régression bayésienne

Julien JACQUES

Préliminaires

Commençons par charger les données

```
x_source <- read.csv("~/Enseignement/Formations/Stat Bayesienne-v2/data IFPEN/dataframe_source.csv")
summary(x_source)
```

```
##        y                X1                X2                X3
##  Min.   :0.7498   Min.   :-21.504   Min.   :-8.0389   Min.   : 5.342
##  1st Qu.:0.8240   1st Qu.:  5.892   1st Qu.:-2.7929   1st Qu.: 7.344
##  Median :0.8283   Median :  9.532   Median :-1.0345   Median : 7.591
##  Mean   :0.8314   Mean   :  8.940   Mean   : 0.3105   Mean   : 7.727
##  3rd Qu.:0.8361   3rd Qu.: 12.075   3rd Qu.: 3.1290   3rd Qu.: 7.920
##  Max.   :0.8873   Max.   : 19.508   Max.   :27.2771   Max.   :14.477
##        X4                X5                X6                X7
##  Min.   :-8.298   Min.   :-18.290   Min.   :-19.716   Min.   :-15.423
##  1st Qu.:-8.242   1st Qu.: -9.381   1st Qu.:  2.102   1st Qu.:-11.476
##  Median :-8.175   Median : -7.427   Median :  3.909   Median : -9.606
##  Mean   :-8.194   Mean   : -5.905   Mean   :  4.496   Mean   : -9.184
##  3rd Qu.:-8.162   3rd Qu.: -2.749   3rd Qu.:  5.764   3rd Qu.: -8.072
##  Max.   :-7.845   Max.   :  9.897   Max.   : 31.801   Max.   : 12.992
##        X8                X9               X10               X11
##  Min.   :-14.645   Min.   :-9.843   Min.   :-9.1910   Min.   :-7.5881
##  1st Qu.: -6.129   1st Qu.: 7.509   1st Qu.:-2.2876   1st Qu.:-1.0621
##  Median : -4.325   Median :10.413   Median :-1.1592   Median :-0.1046
##  Mean   : -3.608   Mean   : 8.741   Mean   :-1.4354   Mean   :-0.1066
##  3rd Qu.: -1.096   3rd Qu.:11.693   3rd Qu.:-0.5475   3rd Qu.: 0.5471
##  Max.   : 14.670   Max.   :14.418   Max.   : 6.9047   Max.   :11.4537
##       X12              PLANT
##  Min.   :-7.0177   Length:3177
##  1st Qu.:-1.0294   Class :character
##  Median : 0.4108   Mode  :character
##  Mean   : 0.4481
##  3rd Qu.: 1.6994
##  Max.   : 6.8346
```

On va ignorer la variable catégoriel PLANT

```
x_source$PLANT=NULL
tmp=scale(x_source[,-1])
x_source[,-1]=tmp
```

Ensuite je réalise une régression linéaire

```
model1=lm(y~.,data=x_source)
summary(model1)
```

```
##
## Call:
## lm(formula = y ~ ., data = x_source)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.078894 -0.002203  0.000020  0.002053  0.020832
##
## Coefficients:
##               Estimate Std. Error   t value Pr(>|t|)
## (Intercept)  8.314e-01  7.011e-05 11858.900  < 2e-16 ***
## X1           1.926e-03  1.177e-04    16.358  < 2e-16 ***
## X2           3.316e-04  1.521e-04     2.180   0.0293 *
## X3           1.001e-03  1.444e-04     6.936 4.86e-12 ***
## X4          -1.094e-03  1.254e-04    -8.722  < 2e-16 ***
## X5           2.835e-03  1.790e-04    15.839  < 2e-16 ***
## X6          -2.584e-03  2.330e-04   -11.092  < 2e-16 ***
## X7           5.222e-04  9.705e-05     5.381 7.94e-08 ***
## X8           1.776e-03  1.540e-04    11.533  < 2e-16 ***
## X9          -1.026e-02  1.391e-04   -73.788  < 2e-16 ***
## X10          8.492e-04  1.789e-04     4.747 2.15e-06 ***
## X11         -9.553e-04  1.598e-04    -5.979 2.49e-09 ***
## X12          9.560e-04  1.493e-04     6.401 1.77e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.003952 on 3164 degrees of freedom
## Multiple R-squared:  0.8626, Adjusted R-squared:  0.8621
## F-statistic:  1655 on 12 and 3164 DF,  p-value: < 2.2e-16
```

On va maintenant réaliser une régression linéaire sur un petit échantillon de données target (on fait comme si on peu de données pour ce nouveau type de catalyseur)

On charge les données

```
x_target <- read.csv("~/Enseignement/Formations/Stat Bayesienne-v2/data IFPEN/dataframe_target.csv")
x_target$PLANT=NULL
x_target[,-1]=scale(x_target[,-1],center = attributes(tmp)$`scaled:center`,scale = attributes(tmp)$`scal
summary(x_target)
```

```
##        y                X1                 X2                X3
##  Min.   :0.8126   Min.   :-5.65826   Min.   :-2.1591   Min.   :-1.12127
##  1st Qu.:0.8246   1st Qu.:-0.27714   1st Qu.:-1.9240   1st Qu.:-0.66559
##  Median :0.8283   Median : 0.01588   Median :-1.8868   Median :-0.52731
##  Mean   :0.8285   Mean   :-0.11069   Mean   :-1.8666   Mean   :-0.53298
##  3rd Qu.:0.8322   3rd Qu.: 0.14241   3rd Qu.:-1.8373   3rd Qu.:-0.41418
##  Max.   :0.8540   Max.   : 0.44864   Max.   :-0.6616   Max.   : 0.04778
##        X4                X5                 X6                X7
##  Min.   : 0.2961   Min.   :-2.6990   Min.   :-2.6554   Min.   :-1.2497
##  1st Qu.: 0.9466   1st Qu.:-0.7271   1st Qu.:-1.2027   1st Qu.: 0.3450
##  Median : 1.1388   Median :-0.6115   Median :-1.0627   Median : 0.7772
##  Mean   : 1.3038   Mean   :-0.6439   Mean   :-1.0053   Mean   : 0.7231
##  3rd Qu.: 1.3163   3rd Qu.:-0.4823   3rd Qu.:-0.8526   3rd Qu.: 1.1619
##  Max.   :26.3655   Max.   : 0.1773   Max.   : 0.3988   Max.   : 2.0597
##        X8                X9                X10               X11
##  Min.   :-2.1469   Min.   :-1.753   Min.   :-1.45107   Min.   :-2.39452
```

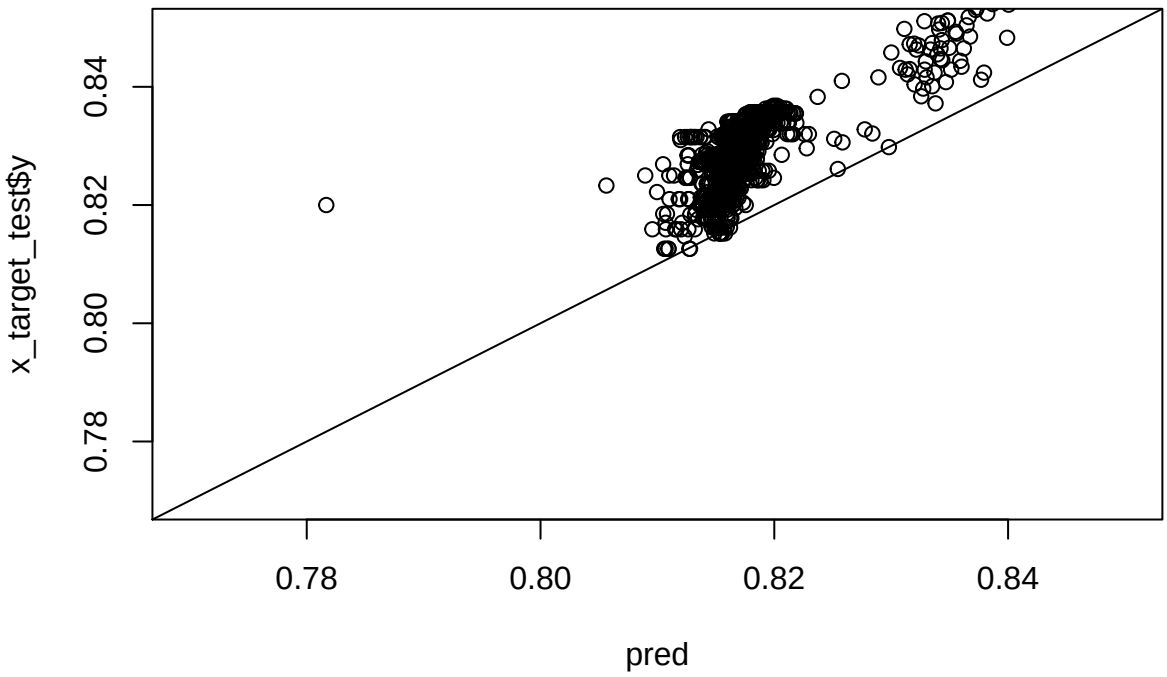
```
##  1st Qu.:-1.1932   1st Qu.: 1.105   1st Qu.:-0.34495   1st Qu.:-1.02814
##  Median :-0.8618   Median : 1.182   Median : 0.09425   Median :-0.83524
##  Mean   :-0.8567   Mean   : 1.030   Mean   : 0.03913   Mean   :-0.82474
##  3rd Qu.:-0.5304   3rd Qu.: 1.231   3rd Qu.: 0.41144   3rd Qu.:-0.62627
##  Max.   : 0.7305   Max.   : 1.466   Max.   : 1.52570   Max.   : 0.03817
##       X12
##  Min.   :-5.6084
##  1st Qu.:-1.2353
##  Median :-1.0804
##  Mean   :-1.0806
##  3rd Qu.:-0.8500
##  Max.   : 0.7334
```

On extrait un échantillon de taille 15 pour apprendre le nouveau modèle, les données restantes nous servirons de test pour évaluer la qualité du modèle

```
set.seed(1111)
indice=sample(1:995,15)
x_target_train=x_target[indice,]
x_target_test=x_target[-indice,]
```

Commençons par évaluer la qualité du modèle estimé sur les données sources

```
pred=predict(model1,x_target_test)
plot(pred,x_target_test$y,xlim = c(0.77,0.85),ylim = c(0.77,0.85))
abline(a=0,b=1)
```




```
cat('MAPE=',sqrt(mean(abs(pred-x_target_test$y)/x_target_test$y)),'\n')
```

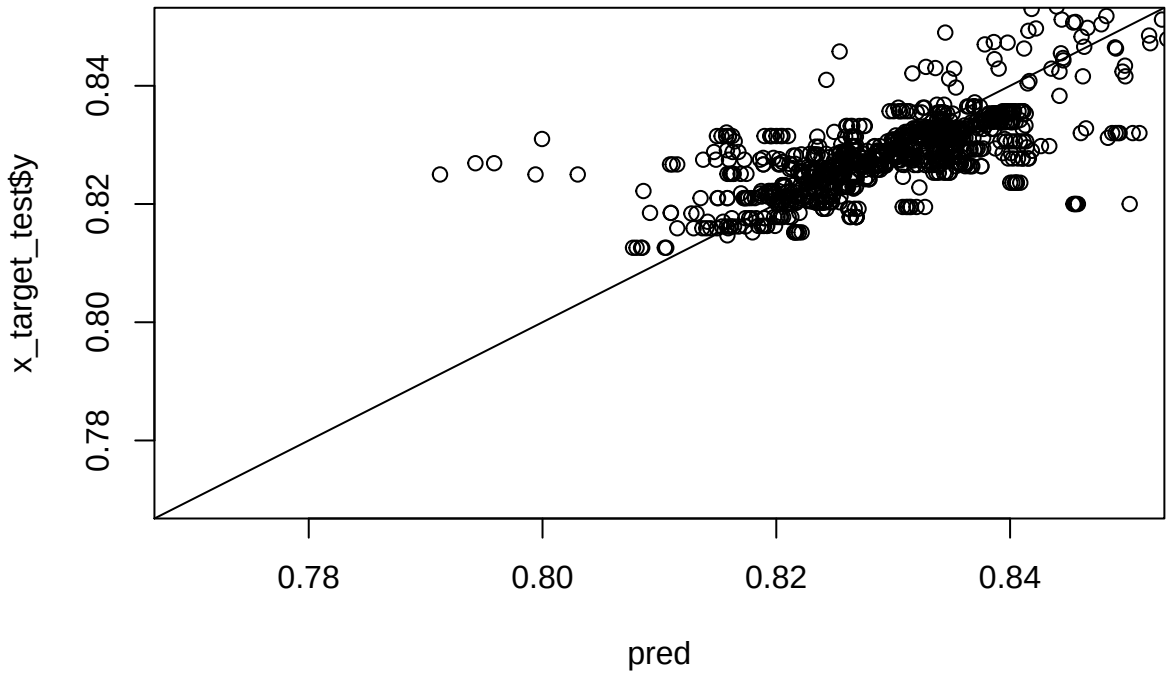
```
## MAPE= 0.1150635
```

Estimons un nouveau modèle sur les données train, de façon fréquentiste

```
model2=lm(y~.,data=x_target_train)
summary(model2)
```

```
##
## Call:
## lm(formula = y ~ ., data = x_target_train)
##
## Residuals:
##        812        438         50        794        292        497        326
##  2.618e-04 -4.162e-04  4.742e-04  1.345e-04  5.009e-05  2.347e-04 -1.525e-04
##        486        986        111        956         63        551        577
## -1.535e-04  8.708e-06 -5.275e-05  1.600e-05 -3.311e-04 -1.783e-04  1.199e-04
##        205
## -1.536e-05
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.8384436  0.0053253 157.445 4.03e-05 ***
## X1           0.0048741  0.0018849   2.586  0.12264
## X2           0.0146982  0.0031284   4.698  0.04244 *
## X3          -0.0033462  0.0026318  -1.271  0.33143
## X4           0.0027488  0.0015250   1.802  0.21326
## X5           0.0124147  0.0027464   4.520  0.04562 *
## X6          -0.0297278  0.0032999  -9.009  0.01210 *
## X7           0.0075779  0.0014008   5.410  0.03251 *
## X8           0.0048807  0.0009888   4.936  0.03868 *
## X9          -0.0122579  0.0013941  -8.792  0.01269 *
## X10          0.0136984  0.0008625  15.882  0.00394 **
## X11         -0.0009642  0.0044503  -0.217  0.84857
## X12         -0.0018182  0.0045171  -0.403  0.72626
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0006117 on 2 degrees of freedom
## Multiple R-squared:  0.9991, Adjusted R-squared:  0.9938
## F-statistic: 188.9 on 12 and 2 DF,  p-value: 0.005278
```

```
pred=predict(model2,x_target_test)
plot(pred,x_target_test$y,xlim = c(0.77,0.85),ylim = c(0.77,0.85))
abline(a=0,b=1)
```




```
cat('MAPE=',sqrt(mean(abs(pred-x_target_test$y)/x_target_test$y)),'\n')
```

```
## MAPE= 0.07330967
```

Cherchons maintenant à faire une régression bayésienne avec comme prior les estimations sur les données sources

```
library(bayess)
```

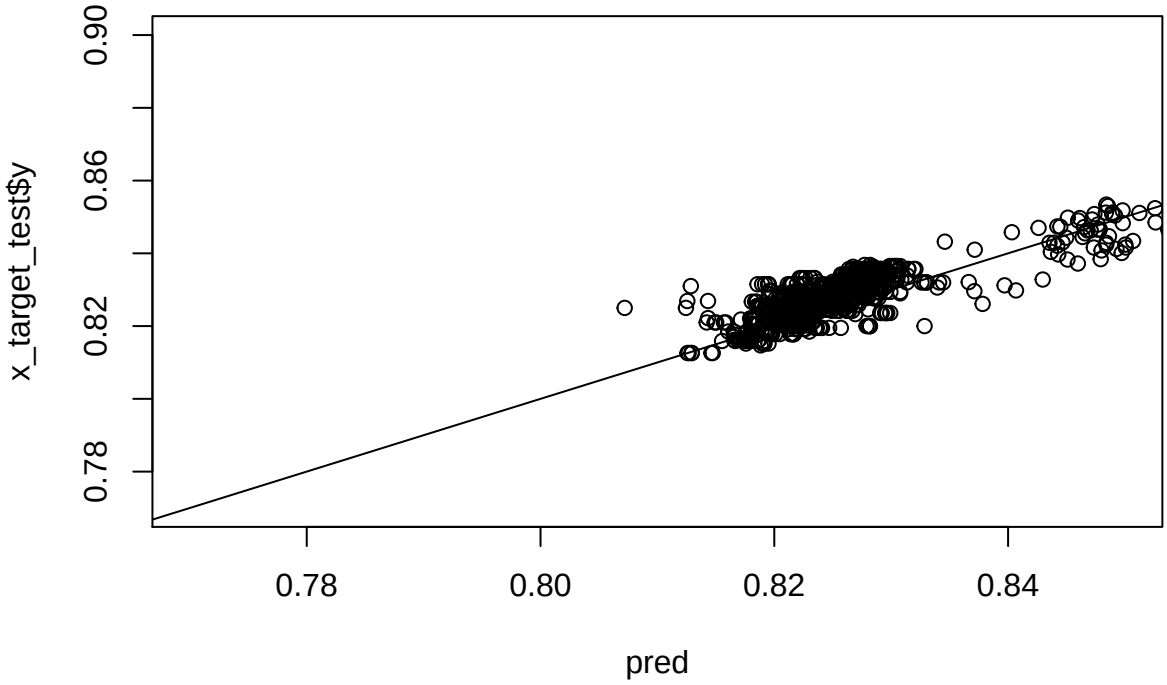
```
## Loading required package: MASS
## Loading required package: mnormt
## Loading required package: gplots
##
## Attaching package: 'gplots'
## The following object is masked from 'package:stats':
##
##     lowess
## Loading required package: combinat
##
## Attaching package: 'combinat'
## The following object is masked from 'package:utils':
##
##     combn
```

```
model3=BayesReg(x_target_train[,1],x_target_train[,-1],betatilde = model1$coefficients[-1],g=10)
```

```
##
##           PostMean PostStError Log10bf EvidAgaH0
## Intercept   0.8295      0.0004
## x1          0.0014      0.0013 -0.2567
## x2          0.0030      0.0017  0.2328       (*)
```

```
## x3         -0.0004      0.0009 -0.5067
## x4          0.0022      0.0033 -0.4121
## x5          0.0043      0.0023  0.2246       (*)
## x6         -0.0104      0.0029   1.671     (***)
## x7          0.0030      0.0014   0.451       (*)
## x8          0.0020      0.0010  0.3568       (*)
## x9         -0.0100      0.0027  1.6811     (***)
## x10         0.0077      0.0012  3.8526    (****)
## x11        -0.0003      0.0025 -0.5205
## x12        -0.0004      0.0030 -0.5189
##
##
## Posterior Mean of Sigma2: 0
## Posterior StError of Sigma2: 0
```

```
pred=model3$postmeancoeff[1]+as.matrix(x_target_test[,-1])%*%model3$postmeancoeff[-1]
plot(pred,x_target_test$y,xlim = c(0.77,0.85),ylim = c(0.77,0.9))
abline(a=0,b=1)
```




```
cat('MAPE=',sqrt(mean(abs(pred-x_target_test$y)/x_target_test$y)),'\n')
```

```
## MAPE= 0.0705599
```

On a effectivement une légère amélioration avec le prior bayésien. Il est démontré dans le papier qu'on peut encore bien améliorer les choses en modifiant l'a priori de Zellner...