Practical Two

February 2018

Detecting collinearity

We will use the same workspace as you generated last week and look at collinearity related issues and 'remedies'.

1. Open your workspace from last weeks practical.
2. Attach the data set.
3. Make pairwise scatter plots for the covariates of interest using the `pairs` function:

```
pairs(cbind(x.pos, y.pos,Depth,impact))
```

4. Check the VIFs for the covariates using the `vif` function and the `glmFitOD2` model.

```
require(car)
vif(glmFitOD2)
```

5. Check the VIFs for the covariates using the `vif` function and the `glmFitOD3` model.

```
require(car)
vif(glmFitOD3)
```

6. Recall the Poisson-like model we are fitting in `glmFitOD3` has:

$$\eta_{it} = \beta_0 + \beta_1 x_{1it} + \beta_2 x_{2it} + \beta_3 x_{3it} + \beta_4 x_{4it} + \beta_5 x_{5it} + \beta_6 x_{6it} + \log(area_{it})$$

$$\mu_{it} = \exp(\eta_{it})$$

and to investigate collinearity, it is prudent to examine potential relationships between the columns of the X matrix (collection of covariates; also called the design matrix).

Examine the columns of the X matrix used to fit the `glmFitOD3` model using the `pairs` function. You can easily extract and view the design matrix using:

```
xmatrix<- model.matrix(glmFitOD3)
head(xmatrix)
```

Ridge regression

1. Fit a ridge regression for the `glmFitOD3` model (and use cross validation to select the tuning parameter (λ)) using the following:

```
require(glmnet)
xmatrix<- model.matrix(glmFitOD3)
ridge<- glmnet(xmatrix, dat$Nhat, family="poisson",
offset=log(area), alpha=0)
cvridge<- cv.glmnet(xmatrix, dat$Nhat, family="poisson",
offset=log(area), alpha=0, nfolds=10)
```

2. Examine these results:

```
par(mfrow=c(1,2))
plot(ridge, xvar="lambda")
abline(v=log(cvridge$lambda.min))
plot(cvridge)
abline(v=log(cvridge$lambda.min))
```

3. Use the following to examine which λ values were trialled and which value was chosen:

```
log(cvridge$lambda)
log(cvridge$lambda.min)
```

4. For clarity, "zoom-in" on the y-axis for the plot using the following:

```
plot(ridge, xvar="lambda", ylim=c(-0.00001, 0.00001))
```

5. Investigate the change in the coefficients for the ridge regression compared to the `glmFitOD3` model:

```
par(mfrow=c(1,1))
cis<- confint(glmFitOD3)
plot(1:7, coef(glmFitOD3), ylim=c(range(cis)),
 xaxt="n", xlab="Coefficients", pch=20)
segments(1:7,cis[,1], 1:7, cis[,2] )
ridgecoefs<- coef(ridge)[-2,
which(ridge$lambda==cvridge$lambda.min)]
points(1:7, ridgecoefs, col=2, pch=20)
axis(1, at=1:7, names(coef(glmFitOD3)))
```

6. Zoom-in for the depth covariate:

```
plot(1:7, coef(glmFitOD3), ylim=c(range(cis[-c(1,3),])), xaxt="n",
 xlab="Coefficients", pch=20)
segments(1:7,cis[,1], 1:7, cis[,2] )
points(1:7, ridgecoefs, col=2, pch=20)
axis(1, at=1:7, names(coef(glmFitOD3)))
```

7. Zoom-in for the `x.pos`, `y.pos` and their interactions:

```
plot(1:7, coef(glmFitOD3), ylim=c(range(cis[-c(1,3,5),])), xaxt="n",
 xlab="Coefficients", pch=20)
segments(1:7,cis[,1], 1:7, cis[,2] )
points(1:7, ridgecoefs, col=2, pch=20)
axis(1, at=1:7, names(coef(glmFitOD3)))
```

8. Are the ridge regression-based coefficients inside the 95% confidence intervals for the `glmFitOD3` model?

```
ifelse(cis[,1]<ridgecoefs & ridgecoefs < cis[,2],1,0)
```

LASSO regression

1.Fit a lasso regression for the `glmFitOD3` model (and use cross validation to select the tuning parameter (λ)) using the following:

```
require(glmnet)
xmatrix<- model.matrix(glmFitOD3)
lasso<- glmnet(xmatrix, dat$Nhat, family="poisson",
```

```
offset=log(area), alpha=1)
cvlasso<- cv.glmnet(xmatrix, dat$Nhat, family="poisson",
offset=log(area), alpha=1, nfolds=10)
```

2. Examine the results:

```
par(mfrow=c(1,2))
plot(lasso, xvar="lambda")
abline(v=log(cvlasso$lambda.min))
plot(cvlasso)
abline(v=log(cvlasso$lambda.min))
```

3. Zoom in for better clarity:

```
par(mfrow=c(1,1))
plot(lasso, xvar="lambda", ylim=c(-0.00001, 0.00001))
```

4. Investigate the change in the coefficients for the lasso regression compared to the `glmFitOD3` model coefficients:

```
par(mfrow=c(1,1))
plot(1:7, coef(glmFitOD3), ylim=c(range(cis)),
 xaxt="n", xlab="Coefficients", pch=20)
segments(1:7,cis[,1], 1:7, cis[,2] )
lassocoefs<- coef(lasso)[-2,
which(lasso$lambda==cvlasso$lambda.min)]
points(1:7, lassocoefs, col=2, pch=20)
axis(1, at=1:7, names(coef(glmFitOD3)))
```

5. Zoom-in for the depth covariate:

```
plot(1:7, coef(glmFitOD3), ylim=c(range(cis[-c(1,3),])),
xaxt="n", xlab="Coefficients", pch=20)
segments(1:7,cis[,1], 1:7, cis[,2] )
points(1:7, lassocoefs, col=2, pch=20)
axis(1, at=1:7, names(coef(glmFitOD3)))
```

6. Zoom-in for x, y and their interactions:

```
plot(1:7, coef(glmFitOD3), ylim=c(range(cis[-c(1,3,5),])),
xaxt="n", xlab="Coefficients", pch=20)
segments(1:7,cis[,1], 1:7, cis[,2] )
points(1:7, lassocoefs, col=2, pch=20)
axis(1, at=1:7, names(coef(glmFitOD3)))
```

7.Examine if the lasso-based coefficients are inside the 95% confidence intervals for the `glmFitOD3` model.

```
lassocoefs
ifelse(cis[,1]<lassocoefs & lassocoefs < cis[,2],1,0)
```

Elastic net regression

1. Fit an elastic net regression for the `glmFitOD3` model (and use cross validation to select the tuning parameter (λ)) using the following:

```
require(glmnet)
xmatrix<- model.matrix(glmFitOD3)
enet<- glmnet(xmatrix, dat$Nhat, family="poisson",
offset=log(area), alpha=0.5)
cvenet<- cv.glmnet(xmatrix, dat$Nhat, family="poisson",
offset=log(area), nfolds=10, alpha=0.5)
```

2. Examine the results:

```
par(mfrow=c(1,2))
plot(enet, xvar="lambda")
abline(v=log(cvenet$lambda.min))
plot(cvenet)
abline(v=log(cvenet$lambda.min))
```

3. Zoom in for better clarity

```
par(mfrow=c(1,1))
plot(enet, xvar="lambda", ylim=c(-0.00001, 0.00001))
```

4. Investigate the change in the coefficients for the elastic net regression compared to the `glmFitOD3` model:

```
par(mfrow=c(1,1))
plot(1:7, coef(glmFitOD3), ylim=c(range(cis)),
 xaxt="n", xlab="Coefficients", pch=20)
segments(1:7,cis[,1], 1:7, cis[,2] )
enetcoefs<- coef(enet)[-2,
which(enet$lambda==cvenet$lambda.min)]
points(1:7, enetcoefs, col=2, pch=20)
axis(1, at=1:7, names(coef(glmFitOD3)))
```

5. Zoom in for the depth covariate:

```
 plot(1:7, coef(glmFitOD3), ylim=c(range(cis[-c(1,3),])),
xaxt="n", xlab="Coefficients", pch=20)
 segments(1:7,cis[,1], 1:7, cis[,2] )
 points(1:7, enetcoefs, col=2, pch=20)
 axis(1, at=1:7, names(coef(glmFitOD3)))
```

6. Zoom in for the x, y and their interactions:

```
 plot(1:7, coef(glmFitOD3), ylim=c(range(cis[-c(1,3,5),])),
xaxt="n", xlab="Coefficients", pch=20)
 segments(1:7,cis[,1], 1:7, cis[,2] )
 points(1:7, enetcoefs, col=2, pch=20)
 axis(1, at=1:7, names(coef(glmFitOD3)))
```

7. Examine if the elastic net-based coefficients are inside the 95% confidence intervals for the `glmFitOD3` model.

```
ifelse(cis[,1]<enetcoefs & enetcoefs < cis[,2],1,0)
```

8. Compare the coefficients for the ridge regression, LASSO and elastic net.

```
lmcoefs<- coef(glmFitOD3)
for(i in 1:length(lmcoefs)){
plot(1:4, c(lmcoefs[i], ridgecoefs[i], lassocoefs[i], enetcoefs[i]),
     ylim=range(c(lmcoefs[i], ridgecoefs[i], lassocoefs[i], enetcoefs[i], cis[i,])),
```

```
     pch=20, col=c(1:4),xlab="Modelling method",
     xaxt="n", ylab="Coefficients", main=names(coef(glmFitOD3))[i])
axis(1, at=1:4, c("LM", "Ridge", "LASSO", "Enet"))
segments(1,cis[i,1],1 , cis[i,2])
abline(h=c(cis[i,1], cis[i,2]))}
```

Questions

1. Which of the following is TRUE?
   - There are no concerns regarding collinearity for the covariates in the `glmFitOD2` model; the pairwise scatterplots show no issues and the VIFs are also very small.
   - There are no concerns regarding collinearity for the covariates in the `glmFitOD2` model; the pairwise scatterplots show no issues and the VIFs are also very large.
   - There are some concerns regarding collinearity for the covariates in the `glmFitOD2` model; the pairwise scatterplots show some issues and the VIFs are also very large.
   - There are some concerns regarding collinearity for the covariates in the `glmFitOD2` model; the pairwise scatterplots show no issues but the VIFs are very small.
   - There are some concerns regarding collinearity for the covariates in the `glmFitOD2` model; the pairwise scatterplots show no issues but the VIFs are very large.

2. Which of the following based on the VIFs for the `glmFitOD3` model is TRUE?
   - The impact covariate and the interaction term containing impact are well explained by the other covariates in the model when each is considered as the response term in a linear model as part of a VIF calculation.
   - The x.pos covariate and the interaction term (containing x.pos) are well explained by the other covariates in the model when each is considered as the response term in a linear model as part of a VIF calculation.
   - The y.pos covariate appears to be highly collinear with the interaction term containing y.pos.
   - No problematic collinearity is evident.
   - Impact, x.pos and y.pos are all collinear with each other.

3. Which of the following is TRUE?
   - The `y.pos:impact` column in the design matrix for `glmFitOD3` always contains zero values (regardless of the `y.pos` value) post-impact.
   - The `x.pos:impact` column in the design matrix for `glmFitOD3` always contains zero values (regardless of the `x.pos` value) post-impact.
   - The `impact` column in the design matrix for `glmFitOD3` always contains zero values (regardless of the `x.pos` value) post-impact.
   - The `x.pos:impact` column in the design matrix for `glmFitOD3` always contains zero values (regardless of the `x.pos` value) pre-impact.
   - The `x.pos:impact` column in the design matrix for `glmFitOD3` is always at least a little bit different to the `x.pos` column to some extent.

4. What is the CV based choice for the tuning parameter for the ridge regression on the log scale? Enter your answer to four decimal places using the same rounding scheme as for Practical 1. 1.3710

5. Which of the following about the ridge regression results is TRUE?
   调整参数
   - The tuning parameter chosen using CV was the largest of those trialled, and so a smaller parameter might still give improved results.
   - The tuning parameter chosen using CV was the smallest of those trialled, and so a larger parameter might still give improved results.
   - The tuning parameter chosen using CV was the smallest of those trialled, and so a smaller parameter might still give improved results.
   - The tuning parameter chosen using CV was the largest of those trialled, and so a larger parameter might still give improved results.
   - The tuning parameter chosen using CV was the smallest of those trialled, and so a smaller parameter is unable to return improved results.

6. Which of the following about the ridge regression results is TRUE?

- The tuning parameter was estimated to be small and so the coefficients are more similar to those based on a GLM, compared to when the parameter is large.
- The tuning parameter was estimated to be large and so the coefficients are more similar to those based on a GLM, compared to when the parameter is small.
- The tuning parameter was estimated to be small and so the coefficients are more different to those based on a GLM, compared to when the parameter is large.
- The tuning parameter was estimated to be large and so the coefficients are more different to those based on a GLM, compared to when the parameter is small.
- When the tuning parameter is extremely large, one or more of the estimated coefficients can be exactly zero.

7. TRUE or FALSE? The estimated coefficient for the x.pos:impact parameter using ridge regression lies inside the 95% confidence interval for this parameter under the GLM.
8. TRUE or FALSE? The estimated coefficient for the y.pos:impact parameter using ridge regression lies inside the 95% confidence interval for this parameter under the GLM.
9. Which of the following about the LASSO regression results is TRUE?

    - One or more of the coefficients can be estimated to be exactly zero (and thus omitted from the model in practice).
    - Only one of the coefficients can be estimated to be exactly zero (and thus omitted from the model in practice).
    - While the tuning parameter can be estimated to be very large, the coefficients can never be estimated to be exactly zero (and thus omitted from the model in practice).
    - In this case, the estimated tuning parameter is very large resulting in coefficients much closer to zero compared with their unpenalised alternatives.

    - The estimated coefficients get closer to zero as the tuning parameter gets closer to zero.

10. TRUE or FALSE? The estimated coefficient for the x.pos:impact parameter using lasso regression lies inside the 95% confidence interval for this parameter under the GLM.
11. TRUE or FALSE? The estimated coefficient for the y.pos:impact parameter using lasso regression lies inside the 95% confidence interval for this parameter under the GLM.
12. Using the point estimate for the CV scores available in each model, which of the models is the 'best'?

lasso