Tutorial 6: Penalised Regression

Ridge regression (with an automatic selection for the ridge parameter) may be carried out using the package **ridge**. You'll also find it in **glmnet**. We'll concentrate on **glmnet**, since it incorporates a larger range of penalised regression methods. The use is reasonably straightforward. Install and activate these packages. The automated parameter selection procedure for the **ridge** package uses a method that may be found in this paper:

`https://arxiv.org/pdf/1205.0686`

In addition to the package **ridge**, the command `lm.ridge` in the package **MASS** also performs a ridge regression.

Least Angle Regression and LASSO can be carried out using routines in the **lars** package and also the **glmnet** package.

The hints in the R file give the 'bare bones'; i.e. the command, the syntax and how to extract minimal information to answer the question. I haven't done this for LASSO using **glmnet**, which you can do for yourselves.

Of course, there is much more information that can be extracted. Please take a while to familiarise yourselves with the options available, what they do and how to extract relevant information.

The R script contains commands for Exercises 1, 2 and 3.

Exercise 1: Polyethylene Data

1. Perform a ridge regression on the `yarn` (or `PET`) data (of course, centring the data first). Consider $k = 0.00001$, 0.01, 0.1 and 1.0. For each of these, plot the coefficient estimate (y-axis) against the coefficient number (x-axis). There are 268 ridge regression coefficients.

   Ridge regression is contained in the **glmnet** package

   ```
   > library(pls)
   > data=yarn
   > library(glmnet)
   Loading required package: Matrix
   Loaded glmnet 4.0-2
   ```

   Now get the ridge parameters and do a ridge regression. This requires `alpha = 0`.

   ```
   > lambdas <- 10^seq(3, -2, by = -.1)
   > fit = glmnet(data$NIR,data$density,alpha=0,lambda=lambdas)
   ```

```
> cv_fit <- cv.glmnet(data$NIR, data$density, alpha = 0, lambda =
lambdas)
Warning message:
Option grouped=FALSE enforced in cv.glmnet, since < 3 observations per
fold
> plot(cv_fit)
```

`cv.glmnet` uses cross validation. This is to obtain the best lambda. We can extract it as follows.

```
> opt_lambda <- cv_fit$lambda.min
> opt_lambda
[1] 0.01
```

The best fit is obtained by

```
fit <- cv_fit$glmnet.fit
```

The fitted values are then:

```
> y_predicted <- predict(fit, s = opt_lambda, newx = data$NIR)
> sst <- sum((data$density - mean(data$density))^2)
> sse <- sum((y_predicted - data$density)^2)
> rsq <- 1 - sse / sst
> rsq
[1] 0.9998843
```

so the modelling accounts for 99.99% of the total sum of squares.

2. Build a model using LASSO. The LASSO procedure involves LARS. How many parameters are included in the model? How does this compare with ridge regression? Which model has the smallest mean squared prediction error?

3. Compare the results from PCR, PLSR (from last tutorial) with ridge and LASSO. Which is most effective for this data set?

Exercise 2: Fitting a LASSO Model

Use the data set `fat` in the library **faraway**. Select the best predictors for body fat (variable `brozek`), using the other variables available, except for `siri`, `density` and `free`.

1. Use the package **glmnet** to fit the linear regression with LASSO.Firstly, define the design matrix of the model X and the outcome Y.

2. Find the best penalty λ using cross validation. The criterion is to minimise the MSE.

3. Investigate the impact of different λs on the coefficients (when $\lambda \to +\infty$ the coefficients go to 0).

4. Compare the estimates using the best value of λ with the estimates obtained using OLS (if the penalisation is low then the results should be similar).

5. What is the predicted percentage of fat (brozek) for someone: age=24, weight=210.25, height=74.75, adipos=26.5, free=167.0, neck=39.0, chest=104.5, abdom=94.4, hip=107.8, thigh=66.0, knee=42.0, ankle=25.6, biceps= 35.7, forearm=30.6, wrist=18.8?

6. Fit the same model above with lasso using the **caret** package.

7. Find the OLS estimates using the **caret** package (you should get the same result). What is the root mean squared error for the lasso and ols models?

Exercise 3: LASSO for a Logistic Model

We deal with logistic regression properly later in the course. For now, you'll see how to use **glmnet** to fit a logistic regression model in the accompanying R script. The data set `bdiag.csv` (on the course page) contains 30 imaging details from patients that had a biopsy to test for breast cancer. The variable diagnosis classifies the biopsied tissue as M for malignant or B for benign.

1. Some variables in the data include measurements that are highly correlated. Compute the correlation between the variables.

2. Some variables, such as `radius_mean` and `perimeter_mean`, have a strong correlation (in this case r = 0.998). What is the difference in the standard error for the coefficient of `radius_mean` when it is fitted alone and when it is fitted together with `perimeter_mean`?

   You should find that the standard error increases almost ten fold, due to collinearity. In more extreme cases, there will numeric problems in the maximisation of the likelihood.

3. Try to fit a logistic regression with all predictors; you should get a message indicating the fitting algorithm did not converge.

4. Use lasso to fit the logistic regression and find λ by cross validation.

   NOTE For logistic regression we do not use mean squared error as the loss function. Instead, we can use another loss function, such as the *deviance* for Bernoulli models. The value of λ which minimises this is used.

5. Assess the logistic model. This may be done using the function `assess.glmnet()`. This gives several statistics, including the area under the ROC curve (c-statistics). The `roc.glmnet()` funtion produces the coordinates for the ROC curve.

   ROC Curve Consider a binary situation with outcomes 1 or 0. A *receiver operating characteristic* curve, or ROC curve, is is created by plotting the *true positive rate* (TPR) against the *false positive rate* (FPR) at various threshold settings. The *true positive rate* (proportion of predicting 1 conditioned the true outcome 1) is also known as the *sensitivity.* The *false positive rate* (proportion of predicting 1s where the true outcome is 0) is $(1 - \text{specificity})$.

   The ROC can also be thought of as a plot of the *power* (accept H_1 when it is true) as a function of the Type I Error (fail to reject H_0 when it is false) of the decision rule.

6. Find the lasso path for the estimates.

Exercise 4: Prostate Data

The dataset `prostate` is available in `faraway`. It contains information on 97 men who were about to receive a radical prostatectomy. These data come from a study examining the correlation between the prostate specific antigen (`logpsa`) and a number of other clinical measures.

Use lasso to fit a linear model and compare the variables selected with backward stepwise regression to predict `logpsa` using all the other predictors.

Exercise 5: Diabetes

Locate the diabetes data in **mlbench**

```
library(mlbench)
#load Pima Indian Diabetes dataset
data(PimaIndiansDiabetes)
```

Use **glmnet** (or otherwise) to perform a LASSO logistic regression, explaining diabetes status (positive or negative) in the last column against the other variables.