Exercises - Analysis of High-Dimensional Data

Nicolas Städler

2024-05-08

Contents

1 Prerequisites

The data sets used in the following exercises can be downloaded from here. We recommend to install the following R packages: `tidyverse`, `knitr`, `caret`, `glmnet`, `MASS`, `lars`, `gbm`, `splines`, `randomForest`, `rpart`, `rpart.plot`, `ggpubr`, `survival`, `survminer`.

Let's load `knitr` and `tidyverse`:

```r
library(tidyverse)
library(knitr)
```

2 Diabetes data and linear regression

The data that we consider consist of 442 diabetes patients, with the response of interest being a quantitative measure of disease progression one year after baseline. There are ten baseline variables — age, sex, body-mass index, average blood pressure, and six blood serum measurements — plus quadratic terms, giving a total of $p = 64$ features.

1. Read the diabetes data set `diabetes.rds` and make a histogram for the response variable `y`. Describe the distribution of the variable.
2. Create a scatterplot matrix for the 5 first variables in the data set. Use `pairs` or `ggpairs`.
3. Randomly assign patients to training and test data (use `sample`).
4. Run a univariate regression model with `bmi` as covariate. Study the `summary` output.
   - How do you interpret the regression coefficients for `bmi`?
   - What is the meaning of the *multiple R-squared*?
   - What is the *residual standard error*?
   - Generate a scatter plot of `y` against `bmi` and add the regression line with confidence band (use `geom_smooth`, `method="lm"`).
   - Draw the Tukey Anscombe plot and the QQ plot (check `?plot.lm`). What are these two plots telling us?
5. Run a multiple regression model using all covariates. Study the `summary` output.
   - What does change in the interpretation of the coefficient for `bmi`?
   - What do you conclude from the *multiple R-squared*?
   - Create a Tukey Anscombe plot and a QQ plot.
6. Calculate the RSS for both models. Write down your observation.
7. Compare the two models using the `anova` function. What do you conclude?

3 Diabetes data and model validation

In the previous section we developed 2 models to predict `y`. In this section we explore the generalizability of these models.

1. Calculate the RMSEs on the training data. Which model will perform better on future data?

2. Use the test data and make scatter plots of the observed against predicted outcomes. Use `ggplot` to create one plot per model and add the regression line (`geom_smooth`) and the "y=x" (`geom_abline`) line to the graph. This plot is also called "calibration plot". The model is "well" calibrated if the regression line agrees with the "y=x" line.
3. Generate boxplots of `predicted - observed` for the 2 models. What do you conclude?
4. Calculate the generalization error, i.e., the RMSE on the test data.

4 Calculus, optimization and OLS

1. Consider the function $f(x) = 2x^2 + x - 5$. Draw a plot of the function.
2. Use `optimize` to find the minimum of $f(x)$.
3. Obtain the minimum of $f(x)$ by taking the derivative and setting equal to zero.
4. Show that $\|a\|_2^2 = a^T a$.
5. Use the result in 4. and show that $\mathbf{RSS}(\beta) = \mathbf{y^T y} - \mathbf{2y^T X}\beta + \beta^{\mathbf{T}}\mathbf{X^T X}\beta$.
6. Invoke the result obtained in 4. and show that

   $$\frac{\partial}{\partial\beta}\mathbf{RSS}(\beta) = -\mathbf{2X^T y} + \mathbf{2X^T X}\beta.$$

   Hint: review the "Identities" section of Matrix calculus - Wikipedia.
7. Do you understand the derivation of the least squares estimator?

5 Diabetes data and regularization

The task is to use the diabetes data to construct a model that predicts the response `y` (disease progression) based on covariates. The two hopes are, that the model would produce accurate baseline predictions of response for future patients, and also that the form of the model would suggest which covariates were important factors in disease progression.

1. Read in the diabetes data set.
2. Run forward stepwise regression. Which is the first variable included in the selection process? Print the coefficients of the AIC-optimal model as a table.
3. Fit Ridge regression. Show the trace plot and the cross-validation plot.
4. Run the Lasso approach and show the trace and the cross-validation plots.
5. Calculate the root-mean-square errors (RMSE) for all 3 models on the test data and compare with the performance of the full model. Which model generalizes best?
6. Plot the regression coefficients for all 3 models.

6 Diabetes data and the `caret` package

In the previous exercise we build predictive models for the Diabetes data. The `caret` package (short for Classification And REgression Training) is a set of functions that attempt to streamline the process for creating predictive models. The aim of this exercise is to get familiar with the `caret` package by running the code below and by investigating the purpose of the functions `train`, `trainControl`. Use the following link to learn more about `caret`.

```r
## Load package
library(caret)

## Read and prepare the data
```

```
set.seed(007)
diabetes <- readRDS(file="data/diabetes.rds")
data <- as.data.frame(cbind(y=diabetes$y,diabetes$x2))
colnames(data) <- gsub(":",".",colnames(data))
train_ind <- sample(seq(nrow(data)),size=nrow(data)/2)
data_train <- data[train_ind,]
xtrain <- as.matrix(data_train[,-1])
ytrain <- data_train[,1]
data_test <- data[-train_ind,]
xtest <- as.matrix(data_test[,-1])
ytest <- data_test[,1]

## Setup trainControl: 10-fold cross-validation
tc <- trainControl(method = "cv", number = 10)

## Ridge
lambda.grid <- fit.ridge.cv$lambda
fit.ridge.caret<-train(x=xtrain,
                       y=ytrain,
                       method = "glmnet",
                       tuneGrid = expand.grid(alpha = 0,
                                              lambda=lambda.grid),
                       trControl = tc
)

# CV curve
plot(fit.ridge.caret)
# Best lambda
fit.ridge.caret$bestTune$lambda
# Model coefficients
coef(fit.ridge.caret$finalModel,fit.ridge.cv$lambda.1se)%>%head
# Make predictions
fit.ridge.caret %>% predict(xtest,s=fit.ridge.cv$lambda.1se)%>%head

## Lasso
lambda.grid <- fit.lasso.cv$lambda
fit.lasso.caret<-train(x=xtrain,
                       y=ytrain,
                       method = "glmnet",
                       tuneGrid = expand.grid(alpha = 1,
                                              lambda=lambda.grid),
                       trControl = tc
)

# CV curve
plot(fit.lasso.caret)
# Best lambda
fit.lasso.caret$bestTune$lambda
# Model coefficients
coef(fit.lasso.caret$finalModel,
     fit.lasso.caret$bestTune$lambda)%>%head
# Make predictions
fit.lasso.caret%>%predict(xtest,
                           s=fit.ridge.cv$lambda.1se)%>%head

## Compare Ridge and Lasso
models <- list(ridge=fit.ridge.caret,lasso=fit.lasso.caret)
resamples(models) %>% summary( metric = "RMSE")
```

7 Closed form solution for Ridge regression

1. Show that the Ridge optimization problem has the closed form solution

$$\hat{\beta}_{\lambda}^{\text{Ridge}} = (\mathbf{X}^{\mathbf{T}}\mathbf{X} + \lambda\mathbf{I})^{-\mathbf{1}}\mathbf{X}^{\mathbf{T}}\mathbf{y}.$$

Hint: calculate the gradient of the loss function $\ell_{\text{Ridge}}(\beta|\mathbf{y},\mathbf{X}) = \text{RSS}(\beta) + \lambda\|\beta\|_2^2$, set equal to zero and solve for β.

2. Use the code below to generate simulated data. Use the formula from the script to calculate the Ridge coefficients for $\lambda = 35$. Compare the coefficients with those obtained using `glmnet`. Hint: Read the following blog on how to scale the λ.

```
set.seed(1)

# simulate data
n <- 20
p <- 15
x <-  matrix(rnorm(n * p), n, p)
y <- x[,1:4]%*%c(2,-2,2,-2)+rnorm(n)
```

8 Bayesian interpretation of Ridge regression (difficult)

1. Write down the log-likelihood of the linear regression model. Note: $Y_i = X_i^T\beta + \epsilon_i$, where $\epsilon_1, \dots, \epsilon_n$ iid $N(0, \sigma^2)$ and $\mathbf{X}$ is a fixed $n \times p$ design matrix.
2. Find the expression for the maximum likelihood estimator.
3. Assuming a prior distribution $\beta_1, \dots, \beta_p$ iid $\sim N(0, \tau^2)$, derive the posterior distribution of β and show that the maximum a posteriori estimator (MAP) coincides with the Ridge estimator.

9 Riboflavin data and elasticnet mixing parameter

1. Load the `hdi` package and load the riboflavin data set `riboflavin.rds`. More information on the data can be obtained using `?riboflavin`.
2. Run the Lasso and generate the trace plot.
3. Run the Elastic net with mixing parameters $\alpha = 0.25, 0.5, 0.75$ and 1 and compare the cross-validation curves. Hint: use the `foldid` argument in `glmnet`.
4. Show the selected genes for the best performing model.

10 Ridge and Lasso for the orthonormal design (difficult)

1. Calculate the Ridge and the Lasso solution for the special case of an orthonormal design matrix.

11 Heart disease data and logistic regression

We explore logistic regression based on the South African heart disease data `sahd.rds`. Proceed as follows:

1. Fit a univariate logistic regression model with `age` as the covariate. Calculate the odds-ratio and elaborate on the interpretation.
2. Predict the probability of heart disease at age 65.

3. Fit a logistic regression model including all covariates. Run stepwise backward selection. Which variables are excluded? What is the AIC value of the final model?
4. Fit a logistic regression model using four natural cubic spline bases for each term in the model. Run backward selection and summarise the final model. Plot the natural cubic spline functions for the age term (use `termplot`). What does the plot tell you?

12 Phoneme recognition

In this exercise we investigate prediction of phonemes based on digitized speech data.

1. Read the data set, subset the phonemes "aa" and "ao" and create training and test data.

```r
dat <- readRDS(file="data/phoneme.rds")
dat2 <- dat[dat$g%in%c("aa","ao"),]

dtrain <- dat2[grepl("^train",dat2$speaker),-c(1,259)]
xtrain <- as.matrix(dtrain[,-257])
ytrain <- ifelse(dtrain$g=="ao",1,0)
dtest <- dat2[grepl("^test",dat2$speaker),-c(1,259)]
xtest <- as.matrix(dtest[,-257])
ytest <- ifelse(dtest$g=="ao",1,0)

dtrain$y <- ytrain
dtest$y <- ytest
dtrain <- dtrain[,-257]
dtest <- dtest[,-257]
```

2. Plot the log-periodogram as a function of frequency for 5 examples each of the phonemes "aa" and "ao".

3. Fit a logistic regression model and evaluate the training and test misclassification errors.

4. Run Lasso regression and evaluate the training and test misclassification errors.

5. In the previous approaches we assumed logit-link

$$\text{logit}(x;\beta) \quad = \quad \sum_{j=1}^{256} X_j \beta_j.$$

Next we assume that the coefficients are a smooth function of the frequency $\beta(f)$, i.e.

$$\beta(f) \quad = \quad \sum_{m=1}^{\nu} h_m(f)\theta_m,$$

where h_m are B-spline basis functions for a natural cubic spline with $\nu = 12$ degrees of freedom (defined on the set of frequencies). Consider filtered inputs $x^* = \mathbf{H}^T x$ and fit θ by logistic regression on the x^*. Evaluate the training and test misclassification errors.

6. Plot the coefficients of the different models.

13 Classification and the `caret` package

Follow the short tutorial which guides you through a classification example using the `caret` package.

14 Survival analysis and the Lymphoma data

In this exercise we explore the Lymphoma data set to predict survival based on gene expression data.

1. Load the Lymphoma data and make a histogram of the survival times (`lymphx.txt` and `lymphtime.txt`) .
2. Plot the estimated survival curve using `survfit` (Kaplan-Meier method).
3. Fit a Cox regression model with the first three genes as predictors. Use the function `coxph`.
4. Build a predictive model using `glmnet` (data pre-processing: use the 100 genes with largest variance and standardize the resulting predictor matrix to have zero-mean and unit-variance). Which genes are selected? What is the C-index for the optimal tuning parameter?
5. Use the predictive model and classify patients into "good" and "poor" prognostic groups by thresholding the linear predictor at zero. Calculate the Kaplan-Meier curves for the two groups. What is your conclusion? Do you have any concerns?
6. The linear predictor scores computed in 5. are biased as they are evaluated on the same data for which they were computed. We now use a variant of cross-validation, known as *pre-validation*, in order to obtain a fair evaluation of the model. Calculate a pre-validated data set using the code below and calculate the Kaplan-Meier curves for patients with good and poor prognosis.

```r
# split data into K=5 folds
n.fold <- 5
foldid <- sample(rep(1:n.fold, length = nrow(x)))

# pre-validation
dat.preval <- data.frame(y)
dat.preval$lp <- NA
```

```
for (i in 1:n.fold){

  # train model on samples not in the kth fold
  omitk <- which(foldid==i)
  fitk <- cv.glmnet(x[-omitk,],y.surv[-omitk,],
                    family="cox",
                    type.measure="C",
                    nfolds = 5,
                    alpha=1)

  # calculated linear predictor on samples in the kth fold
  lp <- predict(fitk,
                newx=x[omitk,],
                s=fitk$lambda.1se,
                type="link")
  dat.preval$lp[omitk] <- lp
}
```

15 Decision trees, Random Forest and AdaBoost

In this exercise we explore decision trees based on the South African heart disease data `sahd.rds`.

1. Load the South African heart disease data and grow a decision tree using `rpart`. Visualize the tree using `rpart.plot`. How many leave nodes has the tree?
2. Re-grow the tree but now relax the "default" control parameters (choose `rpart.control(cp=0, minsplit=50)`). How many leaves has the tree now?
3. Plot the cross-validation error against the complexity parameter α. What is the tree size of the optimal model?
4. Prune the tree using `prune` and by choosing cp ($=\alpha$) to achieve minimal cross-validation error.
5. Calculate the confusion matrix and the misclassification error.
6. Generate a bootstrap sample. Grow a tree and calculate the out-of-bag error.
7. Fit a random forest using `randomForest`. Plot the fitted object. What is this plot telling us? Calculate the variable importance. Which are the most important variables?

16 Email spam and data mining

In this exercise we explore a typical data mining task. We use the `spam.rds` data set. The data for this example consists of information from 4601 email messages. The aim is to predict whether the email is spam. For all 4601 email messages, the true outcome (email type) "email" or "spam" is available, along with the relative frequencies of 57 of the most commonly occurring words and punctuation marks in the email message. This is a supervised learning problem, with the outcome the class variable email/spam. It is also called a classification problem.

1. Read the data set and partition it into 2/3 of training and 1/3 of test data.

2. Use `rpart` to fit a decision tree. Examine the cross-validation output, prune the tree and calculate the misclassification error.

3. Use `randomForest`, calculate the misclassification error and plot the variable importance. How do you check whether the number of fitted trees was sufficient?

4. Use gbm to perform boosting with distribution="bernoulli". How do you set the tuning parameters, i.e. tree size, shrinkage factor and the number of boosting iterations? Calculate the misclassication error and plot the relative influence of the variables.

17 Multiple testing and gene expression

In this exercise we explore the issue of multiple testing using the gene expression data from the mouse experiment (see lecture slides).

1. Load the gene expression data set esetmouse.rds and explore the structure of the object (use the functions str, dim, pData, expr).
2. Carry out a two-sample t-test for a single gene and print the p-value.
3. Repeat 2. for all genes, generate a histogram of p-values and calculated the number of p-values < 0.05.
4. Perform multiple testing correction using the Bonferroni and FDR methods and count the number of significant genes (p.adjust).
5. Use the limma package to run the differential expression analysis and plot the result using a volcano plot (use the functions lmFit, eBayes and volcanoplot)