Regression and Classification Exercises

Soma S Dhavala

```
knitr::opts_chunk$set(echo = TRUE,warning=F,message=F)
set.seed(123)

# UCI ML datasets and many simulated datasets available
require(mlbench)
```

```
## Loading required package: mlbench
```

```
path=getwd()

# no of records
n = 100
# no of feature
p = 10

writeData <- function(X,y,fname)
{
  p <- ncol(X)
  n <- nrow(X)
  print(nrow)
  cnames <- c('target',paste(rep("feature_",p),formatC(1:p,width=floor(1+log10(p)),flag=0,format="d"),s
  # regression problem
  df <- as.data.frame(cbind(y,X))
  colnames(df) <- cnames
  write.csv(df,paste(fname,".regr.csv",sep=""),row.names = F)

  yhat <- y-median(y,na.rm = T)
  prob <- 1/(1+exp(-yhat))
  y <-rbinom(n,1,prob)
  df$target <- y
  write.csv(df,paste(fname,".class.csv",sep=""),row.names = F)
}
```

1. Single Feature

Data

Generate 100 records with 1 features

Comments

Should be straight forward

```
set.seed(1)
X <- matrix(rnorm(n*p),ncol=1)
X <- X^2
beta <- matrix(rnorm(1,0,10),ncol=1)
y <- X%*%beta
writeData(X,y,"Ex01")
```

```
## function (x)
## dim(x)[1L]
## <bytecode: 0x2b555c0>
## <environment: namespace:base>
```

2. Multiple Features

2a. independent features

Data

Generate 100 records with 10 indepdent features.

Comments

- last two features are not important (with lasso, it should produce exact zero)

```
set.seed(2)
X <- matrix(rnorm(n*p),ncol=p)
beta <- matrix(rnorm(p,0,10),ncol=1); beta[c(p-1,p)] <- 0
y <- X%*%beta
writeData(X,y,"Ex02a")
```

```
## function (x)
## dim(x)[1L]
## <bytecode: 0x2b555c0>
## <environment: namespace:base>
```

2b correlated features

Data

Generate 100 records with 10 correlated features.

Comments

- last two features are not important (with lasso, it should produce exact zero)
- gradient descent would be unstable
- variable selection is not consistent

```
set.seed(2)
p=10
n=100
X <- matrix(rnorm(n*p),ncol=p)
X[,3] <- 0.9*X[,1]
X[,p] <- 0.9*X[,1] -0.5*X[,3]
beta <- matrix(rnorm(p,0,10),ncol=1); beta[c(p-1,p)] <- 0
y <- X%*%beta
writeData(X,y,"Ex02b")
```

```
## function (x)
## dim(x)[1L]
## <bytecode: 0x2b555c0>
## <environment: namespace:base>
```

2c features of different scale

Data

Generate 100 records with 10 indepdent features. Each feature is on a different scale and different mean

Comments

- last two features are not important (with lasso, it should produce exact zero)
- gradient descent would be unstable

```
set.seed(2)
p=10
n=100
X <- matrix(rnorm(n*p),ncol=p)
X <- scale(X); X<- scale(X,center=rnorm(p,0,10),scale=abs(0.1+rnorm(p,0.5,15)))
X[,2] <- rnorm(n,1,0.01)
beta <- matrix(rnorm(p,0,10),ncol=1); beta[c(p-1,p)] <- 0
y <- X%*%beta
writeData(X,y,"Ex02c")
```

```
## function (x)
## dim(x)[1L]
## <bytecode: 0x2b555c0>
## <environment: namespace:base>
```

2d correlated features and with different scale

Data

Generate 100 records with 10 correlated features. Each feature is on a different scale and different mean

Comments

- last two features are not important (with lasso, it should produce exact zero)
- gradient descent would be unstable
- variable selection is not consistent

```
set.seed(2)
X <- matrix(rnorm(n*p),ncol=p)
X <- scale(X); X<- scale(X,center=rnorm(p,0,10),scale=abs(0.1+rnorm(p,0.5,15)))
X[,2] <- rnorm(n,1,0.01)
X[,3] <- 0.9*X[,1]
X[,p] <- 0.9*X[,1] -0.5*X[,3]
beta <- matrix(rnorm(p,0,10),ncol=1); beta[c(p-1,p)] <- 0
y <- X%*%beta
writeData(X,y,"Ex02d")
```

```
## function (x)
## dim(x)[1L]
## <bytecode: 0x2b555c0>
## <environment: namespace:base>
```

2e correlated features and with different scale, mising data and outliers.

Data

Generate 100 records with 10 correlated features. Each feature is on a different scale and different mean

Comments

- last two features are not important (with lasso, it should produce exact zero)
- gradient descent would be unstable
- variable selection is not consistent
- regression/classfication is not robust

```
set.seed(2)
X <- matrix(rnorm(n*p),ncol=p)
X <- scale(X); X<- scale(X,center=rnorm(p,0,10),scale=abs(0.1+rnorm(p,0.5,15)))
X[,2] <- rnorm(n,1,0.01)
X[,3] <- 0.9*X[,1]
X[,p] <- 0.9*X[,1] -0.5*X[,3]

# plant missing data
xna.row <- sample(n,5,replace=FALSE)
xna.col <- sample(p,5,replace=TRUE)
X[cbind(xna.row,xna.col)] <- NA

#plan outlier
xna.row <- sample(n,2,replace=FALSE)
xna.col <- sample(p,2,replace=TRUE)
X[cbind(xna.row,xna.col)] <- 1e10
xna.row <- sample(n,2,replace=FALSE)
xna.col <- sample(p,2,replace=TRUE)
X[cbind(xna.row,xna.col)] <- -1e10
beta <- matrix(rnorm(p,0,10),ncol=1); beta[c(p-1,p)] <- 0
y <- X%*%beta
# only in target
yna <- sample(n,2,replace=FALSE)
y[yna] <- NA

writeData(X,y,"Ex02e")
```

```
## function (x)
## dim(x)[1L]
## <bytecode: 0x2b555c0>
## <environment: namespace:base>
```

3 Non-Linear regression

3a Friedman-1 benchmark dataset

Data

Generarte data from

$y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + e$

It has 100 records and 10 features and only five are used

Comments

- can fit linear regression with additional features
- non-parametric method is better in the absence of additional info
- only few features are useful

```
set.seed(3)
xx = mlbench.friedman1(n)
writeData(xx$x,matrix(xx$y,ncol=1),"Ex03a")
```

```
## function (x)
## dim(x)[1L]
## <bytecode: 0x2b555c0>
## <environment: namespace:base>
```

3b Friedman-2 benchmark dataset

Data

Generarte data from

$y = (x_1^2 + (x_2 x_3 - (1/x_2 x_4))^2)^{0.5} + e$

It has 100 records and 4 features

Comments

- non-parametric method is better in the absence of additional info
- linear models will be poor fit

```
set.seed(3)
xx = mlbench.friedman2(n)
writeData(xx$x,matrix(xx$y,ncol=1),"Ex03b")
```

```
## function (x)
## dim(x)[1L]
## <bytecode: 0x2b555c0>
## <environment: namespace:base>
```

4 Ozone Data Set (primarily regression)

Data

Leo Breiman, Department of Statistics, UC Berkeley. Data used in Leo Breiman and Jerome H. Friedman (1985), Estimating optimal transformations for multiple regression and correlation, JASA, 80, pp. 580-598.

Comments

- predict maximum hourly average temperature
- completely exploratory as ground truth is not known

```
set.seed(3)
data(Ozone)
X <- Ozone[,-4]
y <- Ozone[,4]
writeData(X,y,"Ex04")
```

```
## function (x)
## dim(x)[1L]
## <bytecode: 0x2b555c0>
## <environment: namespace:base>
```

5 Satellite Image Data (primarily multi-class classification)

Data

source

Comments

- predict soil type based image pixel values
- completely exploratory as ground truth is not known

```
set.seed(5)
data(Satellite)
write.csv(Satellite,"Ex05.class.csv",row.names = F)
```

Exercises

1. Is a simple linear regression model better choice? Explain in your words what is the functional relationship between the target and the predictor? Can it still be called a linear model?
    - DataSets: 1
    - Miconception: Meaning of Linearity
    - Concepts: run simple linear regression and log-linear model, understand the blackbox, implement simple gradient dsescent and compare model with libraries
2. Is a multiple linear regression model better choice? Explain in your words what is the functional relationship between the target and the predictor?
    - DataSets: >1;
    - Miconception: Meaning of Linearity
    - Concepts: Model Selection, Idea of Baseline Model
3. Comment on the numerical stability of the model fit?
    - DataSets: 2c-2e;
    - Miconception: ML is black-box approach
    - Concepts: Dataset Standarization, Collinearity, Robust regression, Missing Value treatment
4. Is the model explaing the data? Is your model a good model?
    - DataSets: All;
    - Miconception: ML is a black-box approach, I've THE best model
    - Concepts: Model assessment, explainability vs predictive power
5. Is it necessary to preprocess the data? If yes, what sort of data preparation is needed?

- DataSets: >1;
- Miconception: I will be given nice, clean data, all that I need to do is just call a function.
- Concepts: Data cleaning, transformations, check residuals, Iterate between input-model-output-validate
- Methods: Best subset selection (forward, backward, stagewise), lasso, LARS

6. Provide diagnostic plots and critique the model fit (Regression)

   - DataSets: All;
   - Miconception:
   - Concepts: Residual plots, Generalization Error, Test and Train errors, Model fit statistics such as AIC, BIC

   - Techniques/Methods: Cross-Validation, RMSE,

7. Provide diagnostic plots and critique the model fit (Classification)

   - DataSets: All;
   - Miconception:
   - Concepts: Class Imbalance, Multi-class classification, RoC Curve, Classification Truth Table, type-1,2 errors, Classifier summaries
   - Techniques/Methods: Cross-validation, genie-entropy,logistic-regression, Decision-Trees, Resampling

“‘