p8106_hw1_zl2860

Zongchao Liu

2/16/2020

Load data

```
set.seed(886)
train = read_csv("./solubility_train.csv")
test = read_csv("./solubility_test.csv")

train_x = model.matrix(Solubility ~ . , train)[,-1]
train_y = train$Solubility
test_x = model.matrix(Solubility ~ . , test)[,-1]
test_y = test$Solubility
```

1. LS regression

```
set.seed(886)
ctrl1  = trainControl(method = "repeatedcv", number = 10, repeats = 5)
lm.fit = train(train_x,
               train_y,
               method = "lm",
               trControl = ctrl1)

pred_test_lm = predict(lm.fit, test_x)
mse(test_y,pred_test_lm)
```

```
## [1] 0.5558898
```

The mean square error of least square linear regression on the test data is 0.5558898.

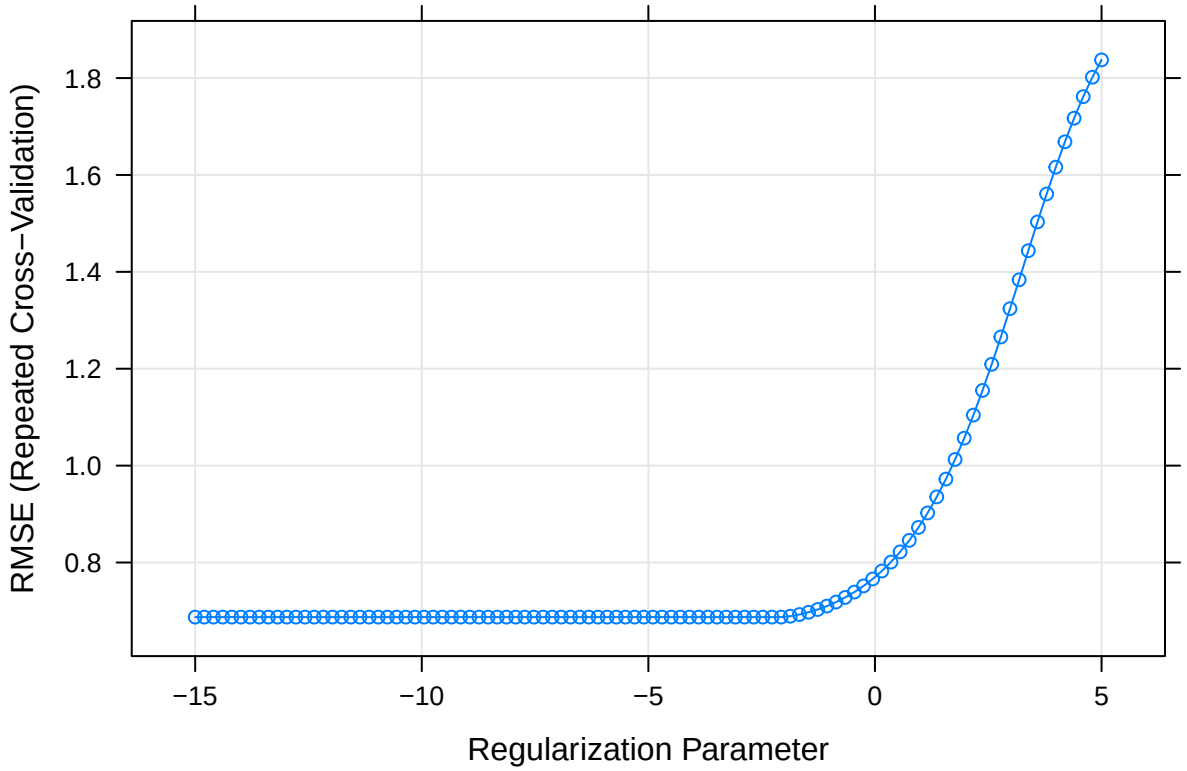
2. ridge

```
set.seed(886)
ridge.fit = train(x = train_x,
                  y = train_y,
                  method = "glmnet",
```

```
                    tuneGrid = expand.grid(alpha = 0,
                                           lambda = exp(seq(-15,5,length = 100))),
                    preProc = c("center","scale"),
                    trControl = ctrl1)

ridge.fit$bestTune #lambda
```

```
##    alpha    lambda
## 65     0 0.1260966
```

```
plot(ridge.fit,xTrans = function(x) log(x))
```




```
pred_test_ridge = predict(ridge.fit, test_x)
mse(test_y,pred_test_ridge)
```

```
## [1] 0.5134603
```

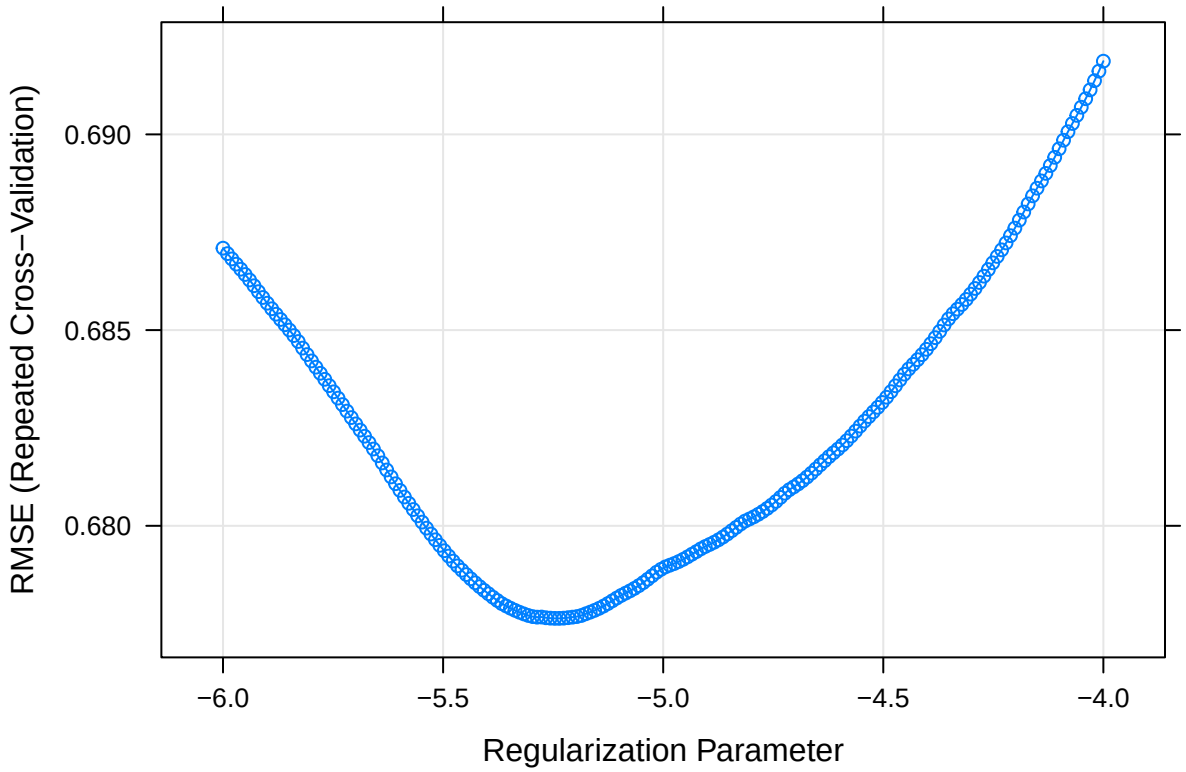
The mean square error of ridge regression on the test data is 0.5134603. The chosen lambda by cross-validation is 0.1260966.

3. Lasso

```
set.seed(886)
lasso.fit = train(train_x,
                  train_y,
                  method = "glmnet",
                  tuneGrid = expand.grid(alpha = 1,
                                         lambda = exp(seq(-6,-4,length = 200))),
                  preProc = c("center","scale"),
                  trControl = ctrl1)


lasso.fit$bestTune
```

```
##    alpha      lambda
## 76     1 0.005267333
```

```
plot(lasso.fit, xTrans = function(x) log(x))
```




```
coef = coef(lasso.fit$finalModel,lasso.fit$bestTune$lambda)
length(which(coef!=0)) # number of non-zero coefficient estimates
```

```
## [1] 144
```

```
pred_test_lasso = predict(lasso.fit,test_x)
mse(test_y,pred_test_lasso)
```

```
## [1] 0.4957342
```

The mean square error of lasso regression on the test data is 0.4957342. The chosen lambda by cross-validation is 0.005267333. The final model using lasso regression has 143 non-zero coefficients and 1 intercept.

d. pcr

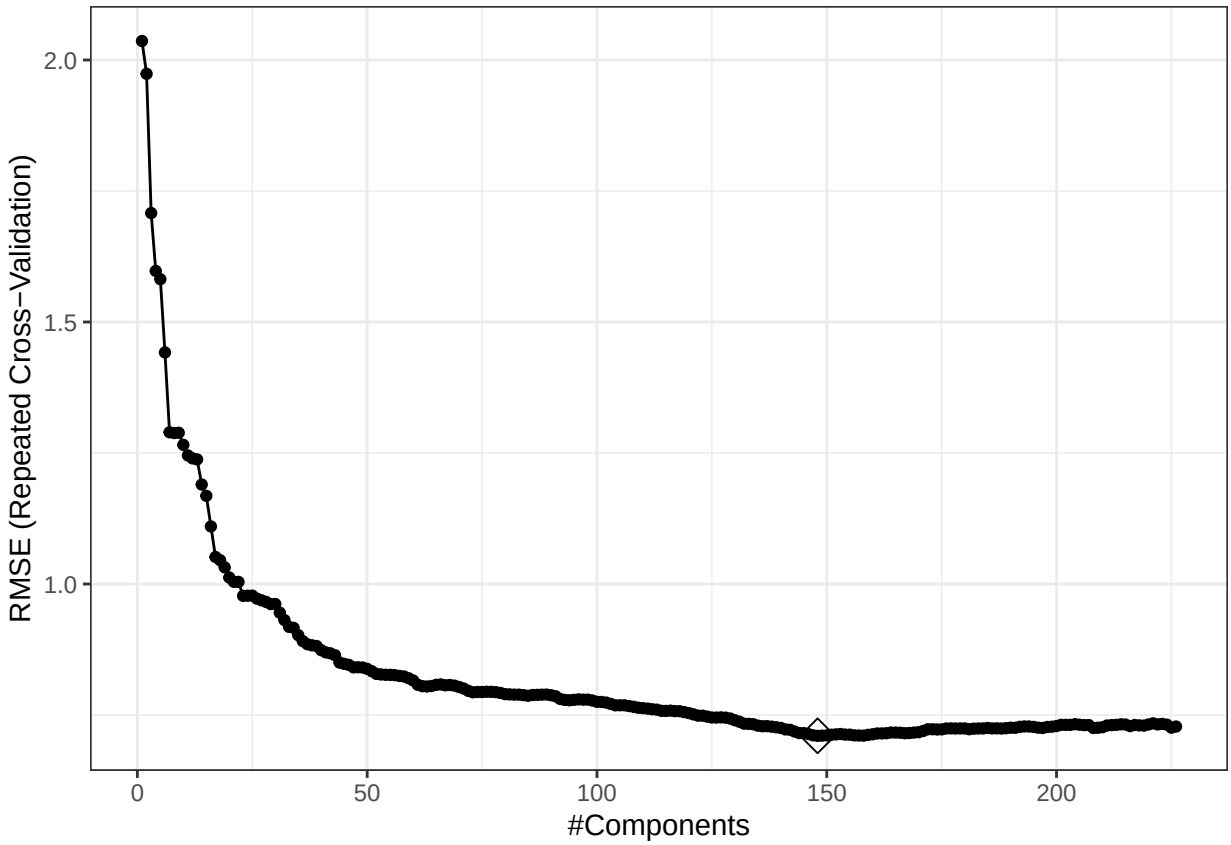
```
set.seed(886)
pcr.fit = train(x = train_x,
                y = train_y,
                method = "pcr",
                tuneGrid = data.frame(ncomp = 1:226),
                tuneLength = length(train),
                trControl = ctrl1,
                preProc = c("center", "scale"))

pred_test_pcr = predict(pcr.fit,test_x)

pcr.fit$bestTune # M=148
```

```
##     ncomp
## 148   148
```

```
mse(test_y, pred_test_pcr)
```

```
## [1] 0.5410365
```

```
ggplot(pcr.fit, highlight = T) + theme_bw()
```



The test error by using crossvalidation is 0.5410365. The value of M selected by cross-validation is 148.

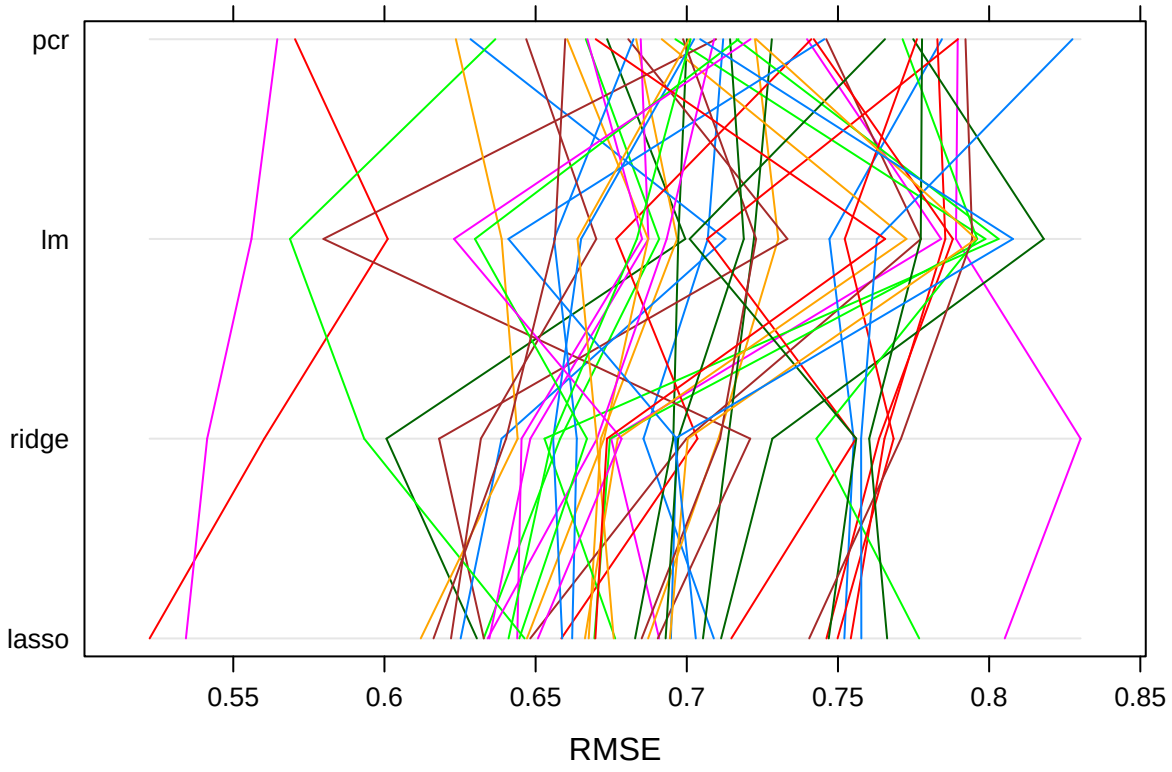
e. briefly discuss the results obtained in a ~ d

```
set.seed(886)
resamp = resamples(list(lasso = lasso.fit,
                        ridge = ridge.fit,
                        pcr = pcr.fit,
                        lm = lm.fit))

summary(resamp)
```

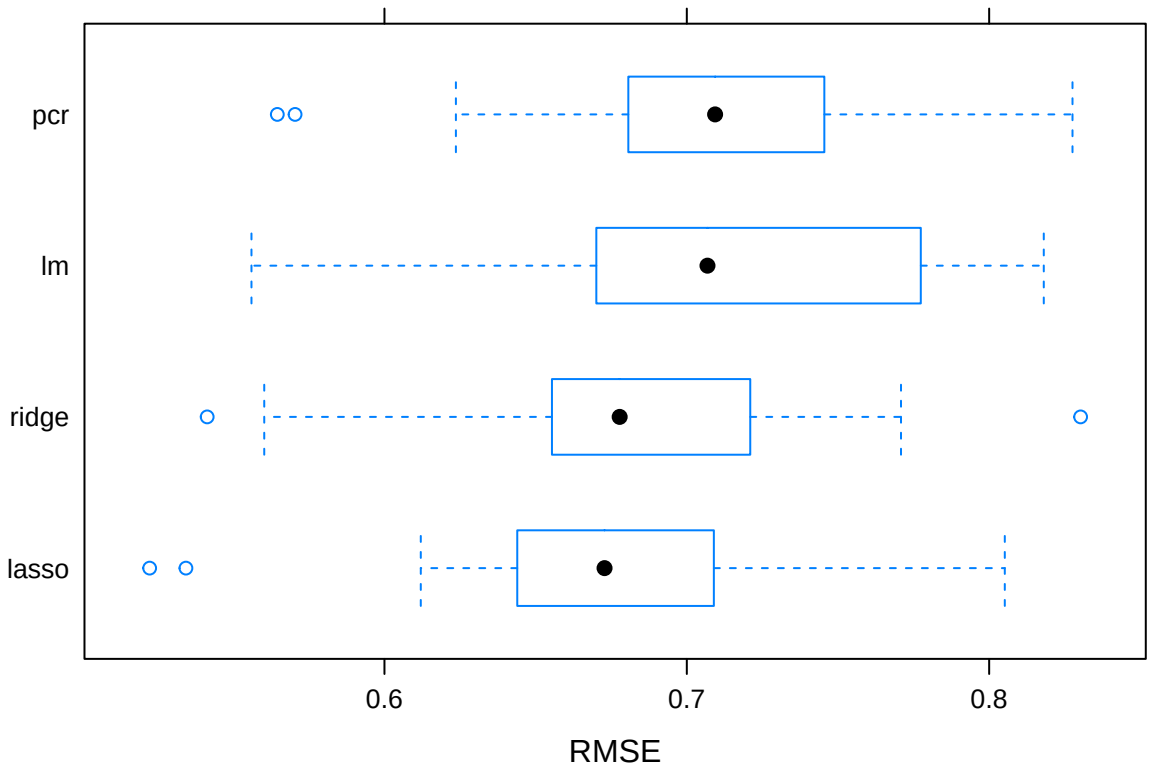
```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: lasso, ridge, pcr, lm
## Number of resamples: 50
##
## MAE
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lasso 0.4115866 0.4900738 0.5143649 0.5187091 0.5507472 0.5932016    0
## ridge 0.4059038 0.5001188 0.5200449 0.5237384 0.5590120 0.6099697    0
## pcr   0.4388351 0.5168060 0.5492244 0.5453486 0.5741605 0.6335138    0
```

```
## lm    0.4223534 0.4993484 0.5260753 0.5317617 0.5669594 0.6223752    0
##
## RMSE
##             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lasso 0.5223360 0.6440894 0.6727906 0.6776355 0.7080456 0.8051749    0
## ridge 0.5413677 0.6554955 0.6777932 0.6872862 0.7191523 0.8302497    0
## pcr   0.5645569 0.6810832 0.7093572 0.7102986 0.7446178 0.8275685    0
## lm    0.5560173 0.6717124 0.7068332 0.7123691 0.7762081 0.8180635    0
##
## Rsquared
##             Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lasso 0.8342757 0.8755701 0.8909826 0.8905513 0.9029722 0.9382580    0
## ridge 0.8291822 0.8694293 0.8913919 0.8874614 0.9016304 0.9381570    0
## pcr   0.8215550 0.8627322 0.8825442 0.8801539 0.8938887 0.9308492    0
## lm    0.8262768 0.8588807 0.8788027 0.8800915 0.8998476 0.9394630    0
```

```
parallelplot(resamp,metric = "RMSE")
```




```
bwplot(resamp, metric = "RMSE")
```



Under the same scenario of cross-validation, by comparing the MSE of each model, we can see that $MSE_{LS} > MSE_{PCR} > MSE_{Ridge} > MSE_{LASSO}$. The LS regression includes all of the predictors to predict the outcome. The ridge and lasso regression model do not use all of the original predictors and instead conduct a feature selection process. For the ridge regression model, the best lambda is 0.1198862. For the lasso regression model, the best lambda is 0.005267333. The principle component model is an unspuervised method to reduce the high dimensions of the data. There is no sample covariance betweendifferent components over the dataset. For the principle component regression model, 148 components are used, which capture 88.02% of the information(variance) of the data.

f. Which model will you choose or predicting solubility?

Based on the resampling results, we can see that lasso regression relatively has the lowest test RMSE. Therefore, for predicting solubility, using lasso regression may be the best choice.