

Lasso-Regression

Carlos Carret Miranda - Faculty of Mathematics and Computation - MATCOM - University of Havana

A prediction problem is carried out, in which a database of mobile phones is given, with several attributes of the same, it is desired to predict the price range that mobile phones can have depending on their characteristics."

First, we import the data; the dimensions of the train and test data will be the same

```
# Loading datasets for training and for testing
attach(train)
attach(test)

# Show dimensions
dim(train)
dim(test)

> dim(train)
[1] 2000  21
> dim(test)
[1] 1009  21
```

As can be seen, we have 2000 data points for training and 1009 data points for testing.

Then, we create a dataset for our independent variables x and another dataset for the dependent variable, which would be the price range.

```
# Getting independent variables.
x=model.matrix(price_range~.,train)

# Removing the intercept
x=model.matrix(price_range~.,train)[,-1]

#Getting dependent variable..
y=train$price_range
```

We run glmnet with alpha=1 to get our Lasso model and analyze it

Fitting a Lasso regression model to the data x and y. The argument alpha = 1 specifies that L1 regularization should be used, which is suitable for feature selection (i.e., identifying which features are important and which are not).

```
lasso.model=glmnet(x,y,alpha =1)
```

Getting the dimensions of the coefficient matrix of the fitted Lasso model. The coefficient matrix contains the coefficients of the independent variables for each lambda value used in the model. The dimensions [1] 21 69 indicate that there are 21 sets of coefficients (one for each lambda value) and 69 independent variables.

```
dim(coef(lasso.model))
```

This line is extracting the coefficient matrix from the fitted Lasso model and storing it in the variable Coeflasso

```
Coeflasso=coef(lasso.model)
```

Printing the coefficient matrix stored in Coeflasso. This allows we to see the coefficients of the independent variables for each lambda value.

```
print(Coeflasso)
```

Accessing the vector of lambda values used in the fitted Lasso model and printing it. The lambda values are important because they determine the level of regularization applied to the model.

```
lasso.model$lambda
```

Creating a plot that shows how the coefficients of the independent variables change with different lambda values. The plot helps to visualize the importance of the features and how regularization affects the model.

```
plot(lasso.model,"lambda",label=TRUE)
```

Searching for the best lambda value with cross-validation

```
cvob.cv=cv.glmnet(x,y,alpha=1)
plot(cvob.cv)
best.lambda =cvob.cv$lambda.min
best.lambda
log(best.lambda)
```

Performing validation with the best lambda value

```
x.test=model.matrix(price_range~,test)[-1]
coef(lasso.model)[,which(lasso.model$lambda==best.lambda)]
pred=predict(lasso.model, s=best.lambda,newx = x.test)
pred
```

Metrics

```
# Root Mean Square Error -- R^2
data.frame(RMSE=RMSE(pred,test$price_range),Rsquare = R2(pred,test$price_range))
```

Finally we obtain:

	RMSE	s1
1	0.3173052	0.9185772

In this case, the RMSE is 0.3173052, which suggests that the model's predictions have an average error of approximately 0.317. The R^2 is 0.9185772, which suggests that the model explains approximately 91.86% of the variability in the price data.