Principal Component Regression (PCR)

Contents

Multiple Linear Regression: Statistical Model

$$Y = b_0 + b_1 X_1 + b_2 X_2 + \ldots + b_p X_p + e$$

Where,

Y	:	Dependent Variable
$X_1, X_2, \ldots, X_p$	:	Independent Variables
$b_0, b_1, \ldots, b_p$	:	Parameters of Model
e	:	Random Error Component

Independent variables can either be **Continuous or Categorical**

Problem of Multicolinearity

Multicolinearity exists if there is a **strong linear relationship among independent variables.**

Consequences

- Highly Unstable Model Parameters
  As standard errors of their estimates are inflated
- Model Fails to predict accurately out of sample Data

Multicolinearity is detected using Variance Inflation Factor, VIF

Tolerance = 1- R_i^2

VIF = 1/Tolerance

where R_i^2 (R Squared) is obtained using regression of Xi on other independent variables

Any VIF > 5, indicates presence of multicollinearity

Multicollinearity – Remedial Measures

The problem of Multicollinearity can be solved by different approaches:

- **Drop one of the independent variables, which is explained by others**
- **Use Principal Component Regression in case of severe Multicollinearity**
- **Use Ridge Regression**

Principal Component Regression

In Principal Component Regression,

First k principal components are used as independent variables instead of original X variables

- Each PC is a linear combination of all X variables
- Final model is expressed in terms of original independent variables for ease of interpretation

Principal Component Regression

Transformation into PCs

The original **p** variables are transformed into a new set of orthogonal or uncorrelated variables called “Principal Components “

Regression Analysis

In the second step, after elimination of the least important principal components, a multiple regression analysis of the response variable against the reduced set of principal components is performed using the OLS estimation

Back Transformation

In the third step, model equation is back transformed in terms of original variables.

PCR-Statistical Model

Model in terms of original X variables:

$$Y = b_0 + b_1x_1 + b_2x_2 + \cdots + b_px_p + e$$

Model in terms of Principal Components:

$$Y = a_0 + a_1PC_1 + a_2PC_2 + \cdots + a_kPC_k + e'$$

Case Study

Background

- A company periodically records data for sales and expenses. The company wishes to model the relationship between its sales and sales related expenses and obtain predictions

Objective

- To predict incremental sales based on planned sales related expenses

Available Information

- **Data available for 143 micro business zones**
- **Sales** is the Dependent Variable
- **Expenditure towards advertisements and promotions in the current and previous months** are Predictors

Data Snapshot

Dependent variable | Independent variables

SRNO	SALES	AD	PRO	SALEXP	ADPRE	PROPRE
1	20.11	1.98	0.9	0.31	2.02	0

Columns	Description	Type	Measurement	Possible values
SRNO	Serial Number	-	-	Intergers
SALES	Incremental Sales	Numerical	INR Million	positive value
AD	Current Advertising Expenses	Numerical	INR Million	positive value
PRO	Current Promotional Expenses	Numerical	INR Million	positive value
SALEXP	Misc. Sales Expenses	Numerical	INR Million	positive value
ADPRE	Previous Period's Advertising Expenses	Numerical	INR Million	positive values
PROPRE	Previous Period's Promotional Expenses	Numerical	INR Million	Positive value

PCR in R

```
# Import csv file "pcrdata"
salesdata<-read.csv("pcrdata.csv",header=T)
```

```
# Fitting a Linear Model :
predsales<-lm(SALES~AD+PRO+SALEXP+ADPRE+PROPRE,data=salesdata)

summary(predsales)
```

- **lm()** fits a linear regression model.
- **summary()** generates model summary.

```
# Output of summary
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -10.8147     6.5314  -1.656  0.10005
AD            4.6762     1.4100   3.316  0.00117 **
PRO           7.7886     1.2628   6.168  7.3e-09 ***
SALEXP       22.4089     0.7704  29.089  < 2e-16 ***
ADPRE         3.1856     1.2442   2.560  0.01154 *
PROPRE        3.4970     1.3697   2.553  0.01177 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.201 on 137 degrees of freedom
Multiple R-squared:  0.9089,    Adjusted R-squared:  0.9055
F-statistic: 273.2 on 5 and 137 DF,  p-value: < 2.2e-16
```

Interpretation: Multiple R-Squared is 0.9089, showing model to be a good fit.

PCR in R

```
# Checking for Multicolinearity
install.packages("car")
library(car)

vif(predsales)
```

vif() in package **car** calculates VIFs.

```
# Output of VIF
       AD       PRO    SALEXP     ADPRE    PROPRE
36.159771 31.846727  1.076284 24.781948 42.346468
```

Interpretation:

- VIF values are very high (>5, except for SALEEXP) indicating severe multicolinearity problem.

PCR in R

```
# PCA in R
# Subsetting data for getting Principal components and performing PCA
salesdatapca<-subset(salesdata,select=c(-SRNO,-SALES))

pc<-princomp(formula=~.,data=salesdatapca, cor=T)
summary(pc)
```

- **princomp()**from base R performs PCA on the
  given numeric data matrix
- **formula=** contains the numeric variables. ~.
  ensures all variables are taken
- **cor=T** indicates that calculations should be done
  using the Correlation Matrix.
- **summary()** generates the summary of PCA

PCR in R

```
# Output
```

```
Importance of components:
                          Comp.1    Comp.2    Comp.3    Comp.4
Standard deviation     1.3015565 1.1318477 1.0705353 0.9334328
Proportion of Variance 0.3388099 0.2562159 0.2292091 0.1742593
Cumulative Proportion  0.3388099 0.5950257 0.8242349 0.9984942
                            Comp.5
Standard deviation     0.086769725
Proportion of Variance 0.001505797
Cumulative Proportion  1.000000000
```

Interpretation:

- The first three principal components explain 82% of the variation in the data. Therefore, we can use 3 PC's in Regression Model

PCR in R

```
# PCR in R
install.packages("pls")
library(pls)
pcmodel<-
pcr(SALES~AD+PRO+SALEXP+ADPRE+PROPRE,ncomp=3,data=salesdata,scale=TRUE)

sales
head(salesdata)
```

Install and load package **pls** (Partial Least Squares).

- **pcr()** in package **pls** performs Principal Component Regression
- **ncomp=3** is the number of components to be included in the model
- **scale=TRUE** indicates X is scaled by dividing each variable by its standard deviation. This ensures data is standardised before running the PCR algorithm.

predict() is used to get predictions by PCR.

PCR in R

```
# Output
```

	SRNO	SALES	AD	PRO	SALEXP	ADPRE	PROPRE	pred_pcr
1	1	20.11	1.98	0.9	0.31	2.02	0.0	21.29053
2	2	15.10	1.94	0.0	0.30	1.99	1.0	18.16976
3	3	18.68	2.20	0.8	0.35	1.93	0.0	21.27149
4	4	16.05	2.00	0.0	0.35	2.20	0.8	17.62114
5	5	21.30	1.69	1.3	0.30	2.00	0.0	22.97930
6	6	17.85	1.74	0.3	0.32	1.69	1.3	20.57217

- **pred_pcr** column gives predicted values of SALES using PCR.

Comparing Linear Regression Model and PCR model on Test data

```
# Importing Test Data
salesdata_test<-read.csv("pcrdata_test.csv",header=TRUE)
```

```
# Getting RMSE of linear regression model
salesdata_test$lmpredict<-predict(predsales,salesdata_test)
salesdata_test$lmres<-(salesdata_test$SALES-salesdata_test$lmpredict)
RMSE_lm<-sqrt(mean(salesdata_test$lmres**2))
```

predict () will give the predicted value for the model.

```
# Getting RMSE of PCR model
salesdata_test$pcrpredict<-predict(pcmodel,salesdata_test,ncomp=3)
salesdata_test$pcrres<-(salesdata_test$SALES-salesdata_test$pcrpredict)
RMSE_pcr<-sqrt(mean(salesdata_test$pcrres**2))
```

Comparing Linear Regression Model and PCR model on Test data

```
# Viewing data after adding predicted & residual variables
```

```
head(salesdata_test)
```

```
# Output
```

	SRNO	SALES	AD	PRO	SALEXP	ADPRE	PROPRE	lmpredict	lmres	pcrpredict	pcrres
1	1	28.93	2.75	1.00	0.72	1.97	0.02	32.31368	-3.3836776	23.23291	5.6970943
2	2	25.96	1.73	1.06	0.89	2.77	0.02	34.36925	-8.4092464	22.26693	3.6930660
3	3	31.25	2.19	1.26	0.79	1.22	0.42	32.29821	-1.0482117	27.61578	3.6342207
4	4	25.05	1.82	1.45	0.83	2.23	0.15	35.21751	-10.1675083	25.21307	-0.1630736
5	5	27.32	2.38	1.01	0.74	1.01	0.07	28.22616	-0.9061594	27.05439	0.2656139
6	6	23.23	2.97	0.46	0.96	2.36	0.12	36.10681	-12.8768143	20.92296	2.3070370

```
RMSE_lm
[1] 9.111682
RMSE_pcr
[1] 2.851245
```

Interpretation:

- **RMSE using PCR is less than RMSE using linear regression**, we may conclude that PCR model predicts SALES better than linear regression model when multicolinearity exists.

Quick Recap

Multiple Linear Regression and Multicollinearity

- Highly correlated predictor variables is a very frequent phenomenon in real world analytics.

Principal Component Regression

- PCR is a three way process where the variables are first transformed to principal components, regression is run by considering these components as regressors and finally, they are transformed back to their original forms.

PCR in R

- `pcr()` function in package `pls` performs PCR.

THANK YOU!