Regression Modeling on Multicollinearity

Richard Feng

May 17, 2019

Abstract

We sought to create a model to predict a person's body fat percentage using easy to obtain bodily measurements. Along the way, we detected multicollinearity in our dataset, and we evaluated different approaches to handling collinearity and produce the best model in predicting body fat percentage. It was found that in terms MSE, our Ridge Regression model resulted in better performance. However using the criteria of R_2, Principal Component Regression resulted in the best model.

1 Introduction

Our project seeks to apply two different regression techniques, principal component regression and Ridge Regression, and evaluating metrics on how well they work in application. Specifically, our dataset had multicollinearity, so we evaluated how well Ridge Regression and PCR handled this situation using different metrics such as R^2 and MSE. We detected the multicollinearity in our dataset through some basic data exploration such as the correlation matrix, and solidified this hunch of collinearity by evaluating the variance inflation factors of the variables.

2 Exploratory Data Analysis

2.1 Dataset

Our dataset consists of 16 variables and 250 observations of people and their corresponding physical attributes such as weight, height and wrist size. The predictor column we want is Pct.BF, which is the percentage of body fat that individual had.

Variable	Definition
Pct.BF	Percentages of body fat
density	Density (gm/cm$\hat{2}$)
age	Age (yrs)
weight	Weight (lbs)
height	Height (inches)
neck	Neck circumference (cm)
chest	Chest circumference (cm)
abdomen	Abdomen circumference (cm)
waist	waist circumference
hip	Hip circumference (cm)
thigh	Thigh circumference (cm)
knee	Knee circumference (cm)
ankle	Ankle circumference (cm)
bicep	Extended biceps circumference (cm)
forearm	Forearm circumference (cm)
wrist	Wrist circumference (cm)

Table 1: Background Information on the Variables

When doing some initial data analysis and testing out some simple linear regression models, we found that density was a very strong indicator of how much body fat a person has. However, measuring density is also very tedious similar to finding an individuals body fat percentage. If we had data on a person's density, then it's very likely that person has the resources to figure out their body fat percentage as well. Thus, since density is not a measurement that is simple to obtain, we removed it from our dataset.

Next, we began exploring the variables and seeing if other variables had relationships with each other using a correlation matrix shown below. Here, we can see that weight seems to have some correlation with other variables such as abdomen and waist (shown in both the height column and height row). Additionally, chest, abdomen, and waist seem to have some strong correlations as well.

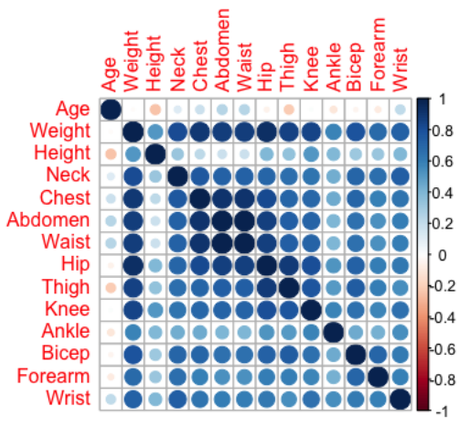

Figure 1: Correlation Matrix

After removing density from our dataset, we scaled our variables since some features were in centimeters while other features such as weight and height are in different units. We then split the dataset into 80% training data, 20% testing data to evaluate our models and how well they could predict an individual's body fat percentage.

3 Statistical Analysis

Below, we conceptually analyze the different statistical methods and metrics used for our results such as variance inflation factor, Principal Component Analysis, Principal Component Regression, and Ridge Regression.

3.1 Proposed Methods

3.1.1 Variance Inflation Factor

After seeing the correlation matrix, we believe that there exist some multicolinearlity issue in our data set. In order to provide a numerical values for the severity of colinearlity in each variable, we need to measure the Variance Inflation Factors (VIF) for each variable. The VIF usually measures the magnitude of multicolinearlity in the variables. Based on the ordinary least square of all variables, we are able to obtain the coefficient of determination R_2 for each variable i and then calculate the corresponding VIF. The R_2 for each variable i is obtained by regressing all the remaining variables on i. The formula to calculate VIF is given below in (1)

$$VIF_i = \frac{1}{1 - R_i^2} \tag{1}$$

A common cutoff criteria for VIF is about 5. If the VIF exceeds 5 for a certain variable, that means there exist some conlinearlity issue among the variable and the remaining variables. If the VIF exceeds 10, it indicates that there exist severe conlinearlity issue. As a

result, the coefficients associated those variables with high VIF are not well estimated based on the regression model.

3.1.2 Principal Component Analysis

Principle Component Analysis allowed us to extract and visualize information in a high dimensional data. The principal components are most commonly used as a means of constructing an informative graphical representation of the data or as input to some other analysis.

The PCs can be obtained by first decomposing the matrix X into USV' where the first n columns of US are the n PCs (Loadings) for the matrix X. The purpose of the Principal Component Analysis here allows us to identify the Principal Components that explaines most of the variance and uses only those PCs as regressors for the following Principal Component Regression.

3.1.3 Principal Component Regression

In order to overcome the multicollinearity issue existed in our dataset, we decide to use the principal components to build a Principal Component Regression model in the previous section. Instead of using the features to build a regression model, PCR uses principal components as the exploratory variables to fit the regression model. In this way, the multicollinearity issue in each variables are not able to affect the model performance since the explanatory variables are the principal components.

The reason that the PCR is able to effectively solve the issue of multiconlinearlity is because each explanatory variable (principal component) which is the column of US is orthogonal to each other. Thus, all of them form a orthogonal basis for the original space.

3.1.4 Ridge Regression

Ridge regression is a well known technique for multiple regression data. What's unique about Ridge Regression compared to other regression techniques such as simple and multiple linear regression is that it's known for its ability to handle data that suffers from multicollinearity. Data that suffers from multicollinearity leads to a Covariance Matrix that does not have full Rank. With this Covariance Matrix, we cannot invert to determine the Least Squares solution; this causes the numerical approximation of the Least Squares Coefficients to blow up to infinity. Ridge Regression introduces the penalty term on the Covariance Matrix to allow for matrix inversion and convergence of the LS Coefficients.

In ordinary least squares regression, the predictions rely on $(X'X)^{-1}$. If $(X'X)$ are nearly singular, then OLS has problems computing the coefficients β. This is a problem we ran into when we tried running a multiple linear regression on our data.

$$J(w) = \lambda w^2 + \sum_i (w^T x_i - y_i)^2. \tag{2}$$

Ridge regression minimizes squared error while regularizing the norm of the weights. To optimize this minimization and our Ridge Regression model, a hyperparameter λ must be tuned, as seen in the equation above. This lambda controls the amount of shrinkage towards 0. However, if lambda is too large, the degrees of freedom will decrease, since the parameters are heavily constrained. On the other hand, if lambda is 0, there is no penalization and we end up using every parameter.

Since Ridge Regression performs L2 regularization, we hope to see a less overfitted, more general model. Ridge regression is less sensitive to outliers and other extreme variance (hence the more general model) which are some properties aids in its success as a model.

3.2 Results

3.2.1 Collinearity Analysis

As in the previous section that the correlation plot suggests that our dataset seems to suffer from some multicolinearlity issues, we need to use statistical methods in order to prove that. First, we proceeds the following metrics as the indicator of the colinearlity. They are the Determinant of correlation matrix, Farrar and Glauber chi-square test, Red indicator,

Sum of reciprocal of Eigenvalues, Theils Method, Condition Number of the data matrix. We obtained the results in the Figure 2 where 1 indicates that the colinearlity is detected and 0 indicates that the colinearlity is not detected.

	results	detection
Determinant	0.000000e+00	1
Farrar Chi-Square	1.202150e+04	1
Red Indicator	6.126824e-01	1
sum of Lambda Invers	7.387462e+14	1
Theil Indicator	1.135047e+00	1
Condition Number	9.027521e+07	1

Figure 2: Collinearity Detection

As we can observe from the figure above, all six metrics shows 1 which means that the variables in our dataset is actually suffers from the multicolinearlity issue.

Then, we further measures the VIF for each of the variable in order to see which variable has colinearlity issue. The cutoff we used here for the detection of colinearlity in VIF is 10. Figure 3 below shows the result of all the variables.

	VIF	detection
Age	2.278191e+00	0
Weight	4.529884e+01	1
Height	3.439587e+00	0
Neck	3.978898e+00	0
Chest	1.071251e+01	1
Abdomen	2.196878e+14	1
Waist	2.196878e+14	1
Hip	1.214625e+01	1
Thigh	7.153711e+00	0
Knee	4.441752e+00	0
Ankle	1.810253e+00	0
Bicep	3.409524e+00	0
Forearm	2.422878e+00	0
Wrist	3.263677e+00	0

Figure 3: Figure: Variable collinearity detection

The result shows that the variable Weight, Chest, Abdomen, Waist and Hip indicates the sign of colinearity since the VIF for them is greater than 10. Among them, the two variables Abdomen and Waist have the same VIF value and they are all significantly larger than the rest. This means that data for these two variables probably has nearly the same value with only differences in scales. After investigation on these two variables, it is not surprisingly to find out that one is only the factor of another by a constant. Also, the Weight variable has relatively high VIF which means that the weight of a person tends to affect other parts of the body.

3.2.2 Ridge Regression

Using cross validation, we an find optimal lambda values for our ridge regression model. As we can see from the figure below, we want the lambda with the least MSE, which is the optimal lambda and can be found at the lowest point in the figure below.

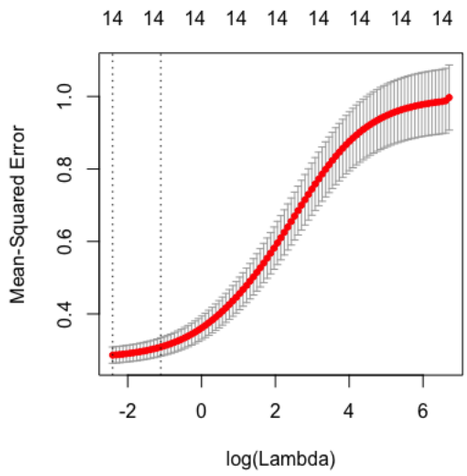

Figure 4: Figure: Log Value of Lambda

Using cross validation, we can see how well the model generalizes using different values of lambda. The k-fold cross validation iterations are summarized in the table below. We can see how although a lower lambda value of 0.08191 was tested out in our cross validation simulation, using 0.0899 achieved a lower MSE and a more general model.

Iteration	DF	%Dev	Lambda
1	14	8.426e-36	819.1
2	14	1.094e-2	746.3
3	14	1.199e-2	619.6
...	...	...	...
97	14	7.374e-1	0.10830
98	14	7.387e-1	0.09866
99	14	7.399e-1	0.08990
100	14	7.410e-1	0.08191

3.2.3 Principal Component Analysis

Thus, we applied the PCA on numerical variables in the dataset except the year of birth because we were interested in the features of companies particularly. In our analysis, the first two principle component explained 61% variance, which was an acceptable percentage Everitt and Hothorn (2011). Thus, it was reasonable to only use the first two principle components.

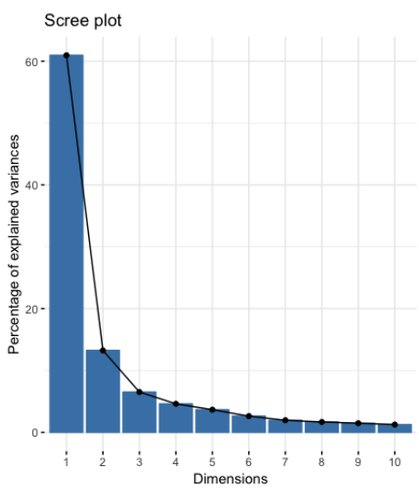

Figure 5: Scree Plot

When observing the biplot below, we can see how density and percentage body fat form a 180°, suggesting there may be a negative correlation between these two. We can also see how height and percentage body fat form a 90°angle, indicating that these two variables may have no correlation with each other. Once again, we see another indication that waist and abdomen may have some collinearity, since they almost overlap each other on the biplot and have some kind of positive correlation. It is interesting to see that many of the variables form small angles, which makes sense as body measurements will be a little related to each other, such as forearm and wrist measurements who both form small angles on the biplot.

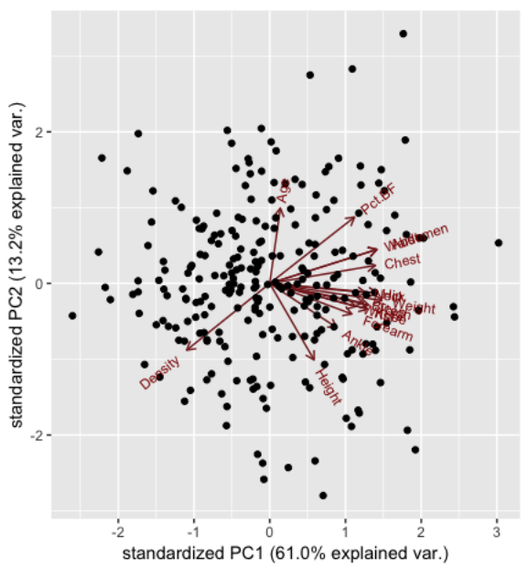

Figure 6: PCA Plot

3.2.4 Principal Component Regression

In order to make our model simple while obtaining reasonable predicting ability, we would like to determine the number of components that should be included included in our model. We firstly use the training data to fit the PCR model with 10-fold cross validation to calculate the mean square error of prediction (MSEP). Then we obtain the MSEP for different number of components in the PCR model by using the validation plot. The plot is shown in Figure 7.

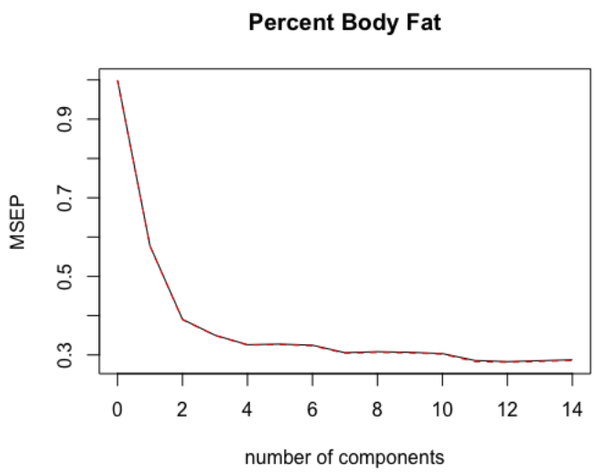

Figure 7: Mean Square Error of Prediction

Based on the validation plot, we are able to observe that after four components, the decrease of MSEP become less apparent. So we decide that a good tradeoff here is to use

only four principal components which explained nearly 85 percent of variance to fit our PCR model. The PCR model we obtained is displayed as following:

$$PCT.BodyFat = 0.0064 + 0.228PC1 + 0.3836PC2 - 0.2375PC3 + 0.1827PC4 \quad (3)$$

Also, the summary of the PCR is shown below:

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.006438   0.028680   0.224    0.823
z1           0.228048   0.009411  24.232  < 2e-16 ***
z2           0.383626   0.021149  18.139  < 2e-16 ***
z3          -0.237537   0.030054  -7.904 1.95e-13 ***
z4           0.182747   0.033954   5.382 2.10e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4056 on 195 degrees of freedom
Multiple R-squared:  0.8379,    Adjusted R-squared:  0.8345
F-statistic: 251.9 on 4 and 195 DF,  p-value: < 2.2e-16
```

Figure 8: Summary of PCR

From the Figure above, all of the four principal components are significant with R^2 equals to 0.8345 indicating this model may performs relatively well. After using the test set to measure the performance of the PCR model, we finally obtain the mean square error of 0.36999.

4 Conclusion

In terms of lowest R_2 value. Principal Component Regression seems to be the superior choice over Ridge Regression in terms of our model to predict body fat percentage.

	RR	PCR
R_2	0.7079	0.8345
MSE	0.2985	0.3600

Using the criteria of lowest mean squared error, Ridge Regression performed better than PCR. Although PCR resulted in a higher R_2 value, the lower MSE from Ridge Regression may create a more general model and handle multicollinearity better across different datasets. The tradeoff between these two ultimately come down to a compromise between accuracy and precision.

To further evaluate the effectiveness of Ridge Regression and PCR handling multicollinearity, similar analysis should be performed on different datasets, such as spatial datasets or datasets on different topics such as genetics to see how well these models and results we found in this paper hold up.

References

Everitt, B. and Hothorn, T. (2011). *An Introduction to Applied Multivariate Analysis with R*. Use R! Springer New York.

Appendix: R code

```
library(mctest)
library(kableExtra)
# library(kable)
data = as.data.frame(scale(data))
x <- data[,-1]
y <- data[,1]
smp_siz = floor(0.8*nrow(data))
set.seed(123)
train_ind = sample(seq_len(nrow(data)),size = smp_siz)
train =data[train_ind,]

x_train = train[,-1]
y_train = train[1]
test= data[-train_ind,]
x_test = test[,-1]
y_test = test[1]

temp <- imcdiag(x,y,vif = 10,method = "VIF")
kable(temp$idiags)
temp <- mctest(x,y,vif = 10)
kable(temp$odiags)

pca = prcomp(train,scale. = T)
pre = predict(pca)
train$pc1 = pre[,1]
train$pc2 = pre[,2]
train$pc3 = pre[,3]
train$pc4 = pre[,4]
pcr.1 = lm(Pct.BF ~ pc1+pc2+pc3+pc4,data = train)
summary(pcr.1)

beta<-coef(pcr.1); A<-loadings(pca)
x.bar<-pca$center; x.sd<-pca$scale
coef<-(beta[2]*A[,1]+ beta[3]*A[,2])/x.sd
beta0 <- beta[1]- sum(x.bar * coef)
c(beta0, coef)

pcr.fit = pcr(Pct.BF~.,data = train, scale = T, validation = "CV")
summary(pcr.fit)
validationplot(pcr.fit,val.type = "MSEP", main = "Percent Body Fat")

pcr.fit4 = pcr(Pct.BF~.,data = train, scale = T, validation = "CV",ncomp = 4)
pcr.pred = predict(pcr.fit4,x_test,ncomp = 4)
mean((pcr.pred-y_test)^2) # Test MSE

cv_fit <- cv.glmnet(x_train, y_train, alpha = 0)
lambda = cv_fit$lambda.min
temp <- cv_fit$glmnet.fit
kable(temp)

#model with optimal lambda
ridge_mod = glmnet(x_train, y_train, lambda = lambda, alpha = 0)
ridge_pred = predict(ridge_mod, s = 10, newx = x_test)
mse = mean((ridge_pred - y_test)^2)

#RR with all variables (including Abdomen + Waist) using training data, with optimal lambda
rmod<-ridge(y_train, x_train, lambda = lambda)
vif(rmod)

lambda = cv_fit$lambda.min
fit <- cv_fit$glmnet.fit
summary(fit)

y_predicted <- predict(fit, s = lambda, newx = x_test)


# Sum of Squares Total and Error
sst <- sum((y_test - mean(y_test))^2)
sse <- sum((y_predicted - y_test)^2)
```

```
mse <- mean((y_predicted - y_test)^2)
rsq <- 1 - sse / sst
rsq
mse
```