PCA is used in exploratory data analysis and for making decisions in predictive models.

PCA commonly used for dimensionality reduction by using each data point onto only the first few principal components (most cases first and second dimensions) to obtain lower-dimensional data while keeping as much of the data’s variation as possible.

The first principal component can equivalently be defined as a direction that maximizes the variance of the projected data.

The principal components are often analyzed by eigendecomposition of the data covariance matrix or singular value decomposition (SVD) of the data matrix.

Reducing the number of variables from a data set naturally leads to inaccuracy, but the trick in the dimensionality reduction is to allow us to make correct decisions based on high accuracy.

Always smaller data sets are easier to explore, visualize, analyze, and faster for machine learning algorithms.

In this tutorial we will make use of iris dataset in R for analysis & interprettion.

**Getting Data**

data("iris")

str(iris)

data.frame’:     150 obs. of  5 variables:

 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...

 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...

 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...

 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...

 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...

In this datasets contains 150 observations with 5 variables.

summary(iris)

Sepal.Length  Sepal.Width   Petal.Length  Petal.Width        Species

 Min.   :4.3   Min.   :2.0   Min.   :1.0   Min.   :0.1   setosa    :50

 1st Qu.:5.1   1st Qu.:2.8   1st Qu.:1.6   1st Qu.:0.3   versicolor:50

 Median :5.8   Median :3.0   Median :4.3   Median :1.3   virginica :50

 Mean   :5.8   Mean   :3.1   Mean   :3.8   Mean   :1.2

 3rd Qu.:6.4   3rd Qu.:3.3   3rd Qu.:5.1   3rd Qu.:1.8

 Max.   :7.9   Max.   :4.4   Max.   :6.9   Max.   :2.5

**Partition Data**

Lets divides the data sets into training dataset and test datasets.

set.seed(111)

ind <- sample(2, nrow(iris),

              replace = TRUE,

              prob = c(0.8, 0.2))

training <- iris[ind==1,]

testing <- iris[ind==2,]

**Scatter Plot & Correlations**

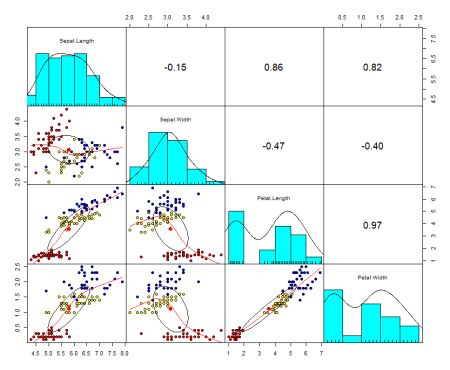
library(psych)

First will check the correlation between independent variables. Let’s remove the factor variable from the dataset for correlation data analysis.

pairs.panels(training[,-5],

             gap = 0,

             bg = c("red", "yellow", "blue")[training$Species],

             pch=21) 

Lower triangles provide scatter plots and upper triangles provide correlation values.

Petal length and petal width are highly correlated. Same way sepal length and petal length , Sepeal length, and petal width are also highly correlated.

This leads to multicollinearity issues. So if we predict the model based on this dataset may be erroneous.

One way handling these kinds of issues is based on PCA.

**Principal Component Analysis**

Principal Component Analysis is based on only independent variables. So we removed the fifth variable from the dataset.

pc <- prcomp(training[,-5],

             center = TRUE,

            scale. = TRUE)

attributes(pc)

$names

[1] "sdev"     "rotation" "center"

[4] "scale"    "x"

$class

[1] "prcomp"

pc$center

Sepal.Length  Sepal.Width Petal.Length

         5.8          3.1          3.6

 Petal.Width

         1.1

Scale function is used for normalization

pc$scale

Sepal.Length  Sepal.Width Petal.Length

        0.82         0.46         1.79

 Petal.Width

        0.76

Print the results stored in pc.

print(pc)

while printing pc you will get standard deviations and loadings.

Standard deviations (1, .., p=4):

[1] 1.72 0.94 0.38 0.14

Rotation (n x k) = (4 x 4):

               PC1    PC2   PC3   PC4

Sepal.Length  0.51 -0.398  0.72  0.23

Sepal.Width  -0.29 -0.913 -0.26 -0.12

Petal.Length  0.58 -0.029 -0.18 -0.80

Petal.Width   0.56 -0.081 -0.62  0.55

For example, PC1 increases when Sepal Length, Petal Length, and Petal Width are increased and it is positively correlated whereas PC1 increases Sepal Width decrease because these values are negatively correlated.

summary(pc)

Importance of components:

                         PC1   PC2    PC3

Standard deviation     1.717 0.940 0.3843

Proportion of Variance 0.737 0.221 0.0369

Cumulative Proportion  0.737 0.958 0.9953

                          PC4

Standard deviation     0.1371

Proportion of Variance 0.0047

Cumulative Proportion  1.0000

The first principal components explain the variability around 73% and its captures the majority of the variability.

In this case, the first two components capture the majority of the variability.

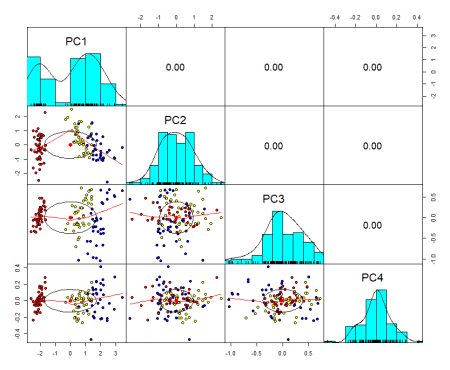
**Orthogonality of PCs**

Let’s create the scatterplot based on PC and see the multicollinearity issue is addressed or not?.

pairs.panels(pc$x,

             gap=0,

             bg = c("red", "yellow", "blue")[training$Species],

             pch=21) 

Now the correlation coefficients are zero, so we can get rid of multicollinearity issues.

**Bi-Plot**

For making biplot need devtools package.

library(devtools)

#install\_github("vqv/ggbiplot")

library(ggbiplot)

g <- ggbiplot(pc,

              obs.scale = 1,

              var.scale = 1,

              groups = training$Species,

              ellipse = TRUE,

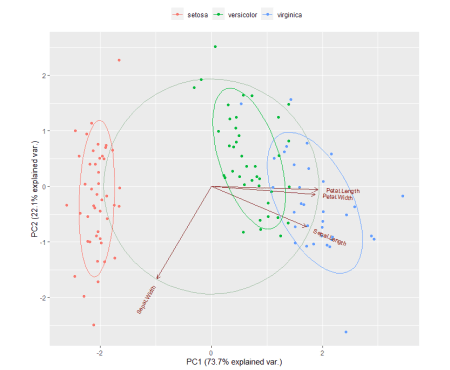
              circle = TRUE,

              ellipse.prob = 0.68)

g <- g + scale\_color\_discrete(name = '')

g <- g + theme(legend.direction = 'horizontal',

               legend.position = 'top')

print(g) 

PC1 explains about 73.7% and PC2 explained about 22.1% of variability.

Arrows are closer to each other indicates the high correlation.

For example correlation between Sepal Width vs other variables is weakly correlated.

Another way interpreting the plot is PC1 is positively correlated with the variables Petal Length, Petal Width, and Sepal Length, and PC1 is negatively correlated with Sepal Width.

PC2 is highly negatively correlated with Sepal Width.

Bi plot is an important tool in PCA to understand what is going on in the dataset.

**Prediction with Principal Components**

trg <- predict(pc, training)

Add the species column information into trg.

trg <- data.frame(trg, training[5])

tst <- predict(pc, testing)

tst <- data.frame(tst, testing[5])

Multinomial Logistic regression with First Two PCs

Because our dependent variable has 3 level, so we will make use of multinomial logistic regression.

library(nnet)

trg$Species <- relevel(trg$Species, ref = "setosa")

mymodel <- multinom(Species~PC1+PC2, data = trg)

summary(mymodel)

Model out is given below and we used only first two principal components, because majority of the information’s available in first components.

multinom(formula = Species ~ PC1 + PC2, data = trg)

Coefficients:

           (Intercept) PC1 PC2

versicolor        7.23  14 3.2

virginica        -0.58  20 3.6

Std. Errors:

           (Intercept) PC1 PC2

versicolor         188 106 128

virginica          188 106 128

Residual Deviance: 36

AIC: 48

**Confusion Matrix & Misclassification Error – training**

p <- predict(mymodel, trg)

tab <- table(p, trg$Species)

tab

Lets see the correct classification and miss classifications in training dataset.

p          setosa versicolor virginica

  setosa         45          0         0

  versicolor      0         35         3

  virginica       0          5        32

1 - sum(diag(tab))/sum(tab)

Misclassification error is 0.067

**Confusion Matrix & Misclassification Error – testing**

p1 <- predict(mymodel, tst)

tab1 <- table(p1, tst$Species)

tab1

Based on the test data set error classification is calculated.

p1           setosa versicolor virginica

  setosa          5          0         0

  versicolor      0          9         3

  virginica       0          1        12

1 - sum(diag(tab1))/sum(tab1)

0.13

Misclassification for the testing data set is 13.33%.