Principal Component Analysis

Jim Harner

1/12/2021

7.1 Principal Component Analysis (PCA)

7.1.1 PCA Basics

Principal Component Analysis (PCA) is used to determine the structure of a multivariate data set composed of numerical variables. Specifically, the most important purposes of PCA are:

1. to reduce the dimensionality of variable space;

2. to find the linear combinations of the original variables which account for most of the variation in the multivariate system.

Let $X_1, X_2, \ldots . X_p$ be numerical variables (or features). The object is to find derived variables, $V_1, V_2, \ldots, V_t$ $(t \leq p)$, such that the V_j are uncorrelated and have successively smaller variances, i.e.,

$$\text{var}(V_1) \geq \text{var}(V_2) \geq \cdots \geq \text{var}(V_t).$$

The V_j are in variable space and are called principal variables. The development is given initially in terms of the sample covariance matrix. Generalizations are then given.

The first principal variable is the linear combination of the X_j with maximum variance. Define

$$V_1 = a_{11}X_1 + a_{21}X_2 + \cdots + a_{p1}X_p = \mathbf{a}_1'X$$

such that $\text{var}(V_1) = \mathbf{a}_1'\mathbf{S}\mathbf{a}_1$ is maximized with respect to $\mathbf{a}_1$. But $\text{var}(V_1)$ can be made arbitrarily large by choosing $\mathbf{a}_1$ such that $\|\mathbf{a}_1\|$ is large. We thus normalize $\mathbf{a}_1$ such that $\|\mathbf{a}_1\|^2 = \mathbf{a}_1'\mathbf{a}_1 = 1$. Thus, the coefficients of V_1 are found from

$$\max_{\mathbf{a}}(\mathbf{a}'\mathbf{S}\mathbf{a})$$

subject to $\mathbf{a}'\mathbf{a} = 1$, or equivalently,

$$\max_{\mathbf{a}}(\frac{\mathbf{a}'\mathbf{S}\mathbf{a}}{\mathbf{a}'\mathbf{a}}).$$

The maximum is l_1, the largest eigenvalue of $\mathbf{S}$. The corresponding normalized eigenvector is $\mathbf{a}_1$. Thus the eigenvector corresponding to the largest eigenvalue determines the first principal variable and

$$\text{var}(V_1) = \mathbf{a}_1'\mathbf{S}\mathbf{a}_1 = l_1.$$

The next problem is to determine the normalized linear combination

$$V_2 = \mathbf{a}_2'X,$$

which has the largest variance in the class of all normalized components orthogonal to V_1 (i.e., constrained by $\mathbf{a}_1'\mathbf{a}_2 = 0$). Geometrically, the axes are perpendicular. The maximum variance is l_2, the second largest eigenvalue of $\mathbf{S}$. The corresponding normalized eigenvector is $\mathbf{a}_2$.

The process can be continued until $t \leq p$ principal variables are found. The j^{th} principal variable is defined by

$$V_j = \mathbf{a}_j' X.$$

Its variance is l_j, the j^{th} largest eigenvalue of $\mathbf{S}$. This last result follows from the eigenvalue problem, since

$$\text{var}(V_j) = \mathbf{a}_j' \mathbf{S} \mathbf{a}_j = l_j \mathbf{a}_j' \mathbf{a}_j = l_j,$$

where $\mathbf{a}_j$ is the normalized eigenvector corresponding to l_j. Also,

$$\text{cov}(V_j, V_k) = \mathbf{a}_j' \mathbf{S} \mathbf{a}_k = 0$$

for $j \neq k$, since $\mathbf{a}_j' \mathbf{S} \mathbf{a}_k = l_k \mathbf{a}_j' \mathbf{a}_k = 0$ by the constraint. Thus, $V_1, V_2, \ldots, V_t$ are uncorrelated and they are ordered by decreasing variability.

PCA reduces analytically to finding the eigenvalue (spectral) decomposition of $\mathbf{S}$ given by:

$$\mathbf{S} = \mathbf{A} \mathbf{D}_{l_j} \mathbf{A}',$$

where the eigenvectors are the columns of $\mathbf{A}$ and the eigenvalues are the diagonal elements of $\mathbf{D}_{l_j}$ (a diagonal matrix). However, the eigenvalue decomposition does not provide the values of the principal variables directly, nor is it the recommended numerical solution. The singular value decomposition is numerically more stable and it provides more information.

The centered data matrix is scaled by $\sqrt{n-1}$ to simplify the interpretation of the subsequent matrix decomposition. Using the "scaled" centered data matrix, $\mathbf{X}_c' \mathbf{X} = \mathbf{S}$. is given by

$$\mathbf{X}_c = \frac{1}{\sqrt{n-1}} (\mathbf{X} - \bar{\mathbf{X}}).$$

The singular value decomposition of the centered data matrix is then:

$$\mathbf{X}_c = \mathbf{V} \mathbf{D}_{d_j} \mathbf{A}',$$

where $d_1 \geq d_2 \geq \cdots \geq d_p \geq 0$ are the singular values, and the columns of $\mathbf{V}$and $\mathbf{A}$ are the left and right singular vectors, respectively.

$$\mathbf{S} = \mathbf{X}_c' \mathbf{X}_c = \mathbf{A} \mathbf{D}_{d_j^2} \mathbf{A}' = \mathbf{A} \mathbf{D}_{l_j} \mathbf{A}',$$

since $\mathbf{V}$ is orthonormal. Thus, the right singular vectors of $\mathbf{X}_c$ are the eigenvectors of $\mathbf{S}$, and the singular values are the square roots of the eigenvalues. Also, the values of the principal variables are given by

$$\mathbf{X}_c \mathbf{A} = \mathbf{V} \mathbf{D}_{d_j},$$

i.e., the j^{th} column of $\mathbf{V} \mathbf{D}_{d_j}$ gives the centered values of the j^{th} principal variable.

Many variants of principal component analysis are possible. The most common is to center and standardize the dataset. Let:

$$\mathbf{X}_s = \frac{1}{\sqrt{n-1}} (\mathbf{X} - \bar{\mathbf{X}}) \mathbf{D}_{1/s_j},$$

where $\mathbf{D}_{1/s_j}$ is diagonal with the reciprocals of the standard deviations of the $Y_j$on the diagonal.

The singular value decomposition of the standardized centered data matrix is given by

$$\mathbf{X}_s = \mathbf{V}_s \mathbf{D}_{d_j^s} \mathbf{A}_s',$$

where the columns of $\mathbf{V}_s$ are the left singular vectors, the columns of $\mathbf{A}_s$ are the right singular vectors, and the d_j^s are the singular values.

Doing a singular value decomposition on $\mathbf{X}_s$ is equivalent to performing an eigenvalue decomposition on the sample correlation matrix $\mathbf{R}$. This follows since $\mathbf{R} = \mathbf{X}_s' \mathbf{X}_s$; the decomposition is

$$\mathbf{R} = \mathbf{A}_s \mathbf{D}_{l_j^s} \mathbf{A}_s',$$

where $l_j^s = (d_j^s)^2$. The sample correlation matrix is the sample covariance matrix of the Y_j^s, i.e, the standardized variables. Note that neither the eigenvalues or eigenvectors of the standardized variables are equivalent to those for the original variables. However, the number of non-zero eigenvalues does not depend on the scaling.

The Spark algorithms are based on the standardized variables. This is done to ensure that the variables are on the same scale.

Geometric Interpretation for Dimension Reduction Principal component analysis can be viewed as a method of fitting subspaces of $\dim t \leq p$ to the data. Consider the case in which $p = 2$. Let the orthogonal distance from $\mathbf{x}_i$ to the coordinate defined by V_1 be d_{i1}. The eigenvector associated with the largest eigenvalue can be found by minimizing $\sum d_{i1}^2$. Notice that d_{i1} is equal to the projection of $\mathbf{x}_i$ onto V_2. Thus,

$$d_{i1}^2 = [\mathbf{a}_2'(\mathbf{x}_i - \bar{\mathbf{x}})]^2.$$

This process can be repeated. In general,

$$d_{it}^2 = \sum_{j=t+1}^{p} [\mathrm{a}_j'[(\mathbf{x}_i - \bar{\mathbf{x}})]^2$$

i.e., d_{it}^2 is the lack-of-fit of the i^{th} individual from the t-dimensional space spanned by $V_1, V_2, \ldots, V_t$.

A t-dimensional subspace may account for most of the variation in a system. Nonetheless, certain $\mathbf{x}_i$ may not lie near this subspace as indicated by large d_{it}^2. Outliers in the $(p-t)$-dimensional space orthogonal to $V_1, V_2, \ldots, V_t$ can be identified by a gamma probability plot. The d_{it}^2 approximately follow a gamma distribution with a shape parameter which must be estimated from the data (i.e., the shape parameter is not $(p-t)/2$).

7.1.2 PCA on the State Crime Data

Read in the crime data for the 50 states:

```
state_crime_df <- read_csv("/home/rstudio/rspark-tutorial/data/state_crime.csv")
```

```
## Parsed with column specification:
## cols(
##   State = col_character(),
##   Abbr = col_character(),
##   Division = col_character(),
##   Region = col_character(),
##   Murder = col_double(),
##   Rape = col_double(),
##   Robbery = col_double(),
##   Assault = col_double(),
##   Burglary = col_double(),
##   Larceny = col_double(),
##   Auto = col_double(),
##   Unemploy = col_double(),
##   Police = col_double(),
##   InSchool = col_double()
## )
```

We remove all variables except the those recording crime per 100,000 residents. These variables are then standardized using the R function `scale`.

```
state_crime_std_df <- state_crime_df %>%
  select(-State, -Abbr, -Division, -Region, -Unemploy, -Police, -InSchool) %>%
  lapply(function(e) scale(e)) %>%
  as.data.frame()
```

The PCA is done on the standardized variables (equivalent to a PCA on the correlation matrix).

```
state_crime_pca <- princomp(state_crime_std_df) %>%
  print()
```

```
## Call:
## princomp(x = state_crime_std_df)
##
## Standard deviations:
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7
## 2.0056558 1.0360906 0.8209734 0.7131056 0.4936528 0.4837463 0.3219325
##
##  7  variables and  50 observations.
```

The standard deviation of the principal variables (the singular values) are extracted and the cumulative percentage of the variability explained is printed.

```
state_crime_var <- state_crime_pca$sdev^2
cumsum(state_crime_var)/sum(state_crime_var)
```

```
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7
## 0.5863929 0.7428774 0.8411278 0.9152560 0.9507797 0.9848921 1.0000000
```

The variable loadings, i.e., the variable coefficients, for the first four principal variables are extracted.

```
state_crime_pca$loadings[, 1:4]
```

```
##             Comp.1      Comp.2     Comp.3      Comp.4
## Murder   0.3915092  0.25913283  0.4100165  0.42283113
## Rape     0.2878928 -0.47640042  0.5987288 -0.55627499
## Robbery  0.4039833  0.42894076 -0.1377602 -0.23858668
## Assault  0.4348569  0.04457428  0.1957617  0.22216445
## Burglary 0.4198884 -0.22326202 -0.1713752  0.39288523
## Larceny  0.2905072 -0.61708392 -0.4931286  0.07516892
## Auto     0.3883742  0.29879303 -0.3788990 -0.49546155
```

The centers and scores can also be extracted for the first four principal variables.

```
head(as.data.frame(state_crime_pca$scores))
```

```
##      Comp.1      Comp.2     Comp.3      Comp.4      Comp.5      Comp.6
## 1 0.4412756  0.68258562  0.8612587  1.20177635 -0.01393582 -0.48753506
## 2 0.4376927 -1.12047545  1.4073123 -1.37994045 -0.03896640  0.04660478
## 3 2.3302987 -1.34645336 -1.7713244  0.23707118 -0.17837232  0.24345638
## 4 0.0857528 -0.07265483  0.9534572  0.79068759  0.01798356  0.25777848
## 5 3.1053344  1.40396903 -0.2582601 -0.38962191 -0.16646047 -0.32057659
## 6 0.1673969 -1.44846124 -0.4895331 -0.05700498 -0.15895049 -0.44643816
##        Comp.7
## 1  0.36692953
## 2  0.68612408
## 3  0.62867914
## 4 -0.02193844
## 5  0.32816975
```

```
## 6  0.08471374
```

7.1.3 Spark PCA on the State Crime Data

Load `state_crime.csv` into Spark with `spark_read_csv` from the local filesystem.

```
state_crime_sdf <- spark_read_csv(sc, "state_crime_sdf",
    path =  "file:///home/rstudio/rspark-tutorial/data/state_crime.csv")
```

The crime rates per 100,000 are extracted for each state.

```
state_crime_std_sdf <- state_crime_sdf %>%
  select(-State, -Abbr, -Division, -Region, -Unemploy, -Police, -InSchool) %>%
  spark_apply(function(e) scale(e))
```

The Spark PCA (`ml_pca`) is run.

```
state_crime_pca_model <- ml_pca(state_crime_std_sdf, k = 4)
class(state_crime_pca_model)
```

```
## [1] "ml_model_pca" "ml_model"
```

The eigenvalues (squares of the singular values) are the variances of the principal variables. The rotation matrix specifies the component loadings.

The cumulative sums estimate the variance explained by the first k principal variables, $k = 1, 2, \ldots, p$.

```
cumsum(state_crime_pca_model$explained_variance)
```

```
##       PC1       PC2       PC3       PC4
## 0.5863929 0.7428774 0.8411278 0.9152560
```

The first two principal variables explain 74.3% of the variation, whereas the first three explain 84.1% of the variation. Thus, we have reduced the dimensionality from $p = 7$ to 3 or 4 dimensions, but with some loss of variation.

We next want to project the data orthogonally into the fitted hyperplane.

```
state_crime_pca_model$pc
```

```
##                 PC1         PC2        PC3         PC4
## Murder   -0.3915092  0.25913283  0.4100165 -0.42283113
## Rape     -0.2878928 -0.47640042  0.5987288  0.55627499
## Robbery  -0.4039833  0.42894076 -0.1377602  0.23858668
## Assault  -0.4348569  0.04457428  0.1957617 -0.22216445
## Burglary -0.4198884 -0.22326202 -0.1713752 -0.39288523
## Larceny  -0.2905072 -0.61708392 -0.4931286 -0.07516892
## Auto     -0.3883742  0.29879303 -0.3788990  0.49546155
```

```
state_crime_pca_proj <- sdf_project(state_crime_pca_model, state_crime_std_sdf)
state_crime_pca_proj
```

```
## # Source: spark<?> [?? x 11]
##    Murder    Rape Robbery Assault Burglary Larceny   Auto     PC1     PC2
##     <dbl>   <dbl>   <dbl>   <dbl>    <dbl>   <dbl>  <dbl>   <dbl>   <dbl>
##  1  1.16  -0.441  -0.0912   1.09   0.00465  -0.493 -0.475 -0.441   0.683
##  2  0.114  2.21   -0.625    0.203 -0.588     0.118  0.345 -0.438  -1.12
##  3  0.165  0.0888  0.0443   0.642  1.61      2.39   1.47  -2.33   -1.35
##  4  0.830  0.221  -0.338    0.191  0.311    -0.376 -0.697 -0.0858 -0.0727
##  5  1.24   0.221   1.77     1.63   0.691     0.162  2.05  -3.11    1.40
##  6 -0.730  0.420  -0.514    0.307  0.305     1.19  -0.172 -0.167  -1.45
```

```
##  7 -0.500 -0.773   0.634   -0.266  0.359    -0.317  0.973 -0.159   0.905
##  8 -0.525  3.20    0.0762   0.376 -0.373     0.300 -0.112 -0.798  -1.75
##  9  0.933  0.818   2.09     2.50   3.03      2.19   1.33  -4.96   -0.774
## 10  1.21   0.950   0.858    0.550  1.47      0.926  0.757 -2.51   -0.419
## # ... with more rows, and 2 more variables: PC3 <dbl>, PC4 <dbl>
```

The `sdf_project` function gives the PCA scores, which than can be used as reduced-dimension features.

```
spark_disconnect(sc)
```