Unsupervised learning

2: principal Component Analysis

Antonio D'Ambrosio

Outline I

Introduction I

Unsupervised learning is often performed as part of an exploratory data analysis.

It can be hard to assess the results obtained from unsupervised learning methods: we have not a "supervisor". In unsupervised learning, there is no way to check our work because we don't know the true answer.

There are several models and methods dealing with unsupervised learning: factor analysis, principal component analysis, correspondence analysis, multidimensional scaling, cluster analysis....

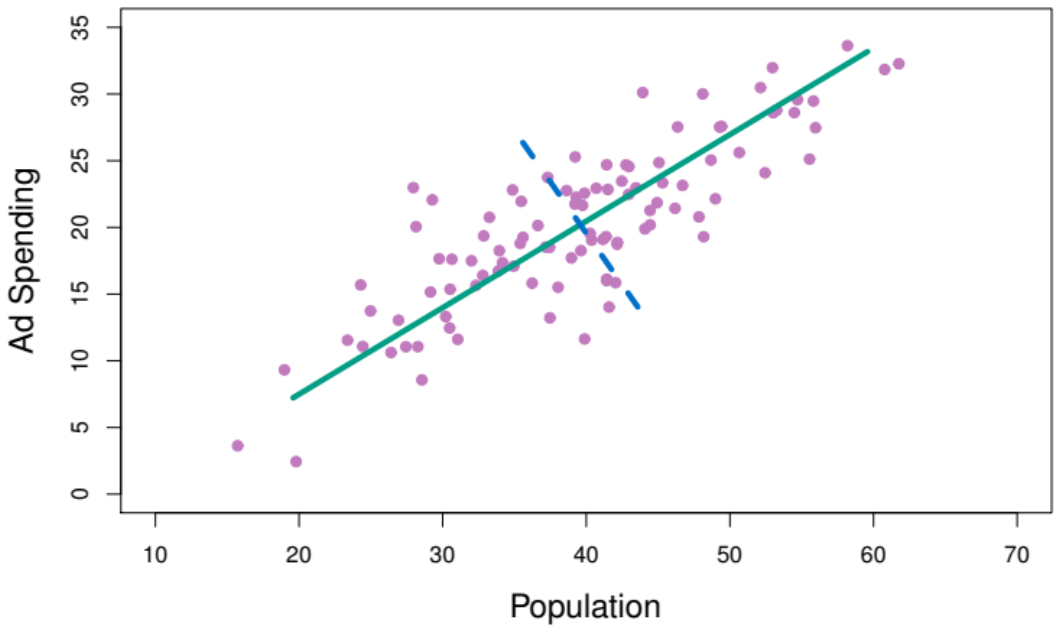
Introduction II

We will focus our attention on Principal Component Analysis.

Principal Component Analysis

Principal Component Analysis (PCA)

- produces a low-dimensional representation of a dataset. It finds a sequence of linear combinations of the variables that have maximal variance, and are mutually uncorrelated.
- Apart from producing derived variables for use in supervised learning problems (PCR, do you remember?), PCA also serves as a tool for data visualization.



The population size (pop) and ad spending (ad) for 100 different cities are shown as purple circles. The green solid line indicates the first principal component direction, and the blue dashed line indicates the second principal component direction.

PCA: details I

Let $\mathbf{X}$ be a $n \times p$ numerical data matrix, with n statistical units and p features.

The first principal component is the linear combination of the features

$$\mathbf{Z}_1 = \mathbf{X}\boldsymbol{\phi}_1 = \phi_{1,1}X_1 + \phi_{2,1}X_2 + \ldots + \phi_{p,1}X_p$$

sub condition that $\boldsymbol{\phi}_1^T \boldsymbol{\phi}_1 = \sum_{j=1}^{p} \phi_{j,1}^2 = 1$.

$\boldsymbol{\phi}_1 = (\phi_{1,1}, \phi_{2,1}, \ldots, \phi_{p,1})^T$ is called principal component loading vector, whose elements are the called loadings of the first principal component.

We have to maximize the quantity $\boldsymbol{\phi}_1^T \mathbf{X}^T \mathbf{X} \boldsymbol{\phi}_1$.

PCA: details II

We always center the variables to have 0 mean. The problem is:

$$\max(\phi_1^T \mathbf{X}^T \mathbf{X} \phi_1) \text{ subject to } \phi_1^T \phi_1 = 1$$

. We have:

$$L = \phi_1^T \mathbf{X}^T \mathbf{X} \phi_1 - \lambda_1(\phi_1^T \phi_1 - 1) = max$$

$$\frac{\partial}{\partial \phi_1} = 2\mathbf{X}^T \mathbf{X} - 2\lambda_1 \phi_1 = 0$$

$$\mathbf{X}^T \mathbf{X} \phi_1 = \lambda_1 \phi_1$$

PCA: details III

- from $\mathbf{X}^T\mathbf{X}\phi_1 = \lambda_1\phi_1$ we see that $\lambda_1 = \phi_1^T\mathbf{X}^T\mathbf{X}\phi_1$, hence this quantity is the sum of squares of the first principal component.
- as the variables are centered, also the first principal component has 0 mean, hence λ_1 is the variance of the first component.
- λ_1 is one of the *eigenvalues* of the matrix $\mathbf{X}^T\mathbf{X}$, hence ϕ_1 is the corresponding *eigenvector*
- what is $\mathbf{X}^T\mathbf{X}$? with centered variables, this matrix corresponds to the covariance matrix of $\mathbf{X}$ (upon a scaling factor equal to $\sqrt{n}$). With standardized variables, it is the correlation matrix.

PCA: details IV

- given that $\mathbf{X}^T\mathbf{X}$ is a squared $p \times p$ (semi-positive defined) matrix, we can compute p eigenvalues and p eigenvectors. The trace of $\mathbf{X}^T\mathbf{X}$ is equal to the sum of the variances of all the p variables.
- it can be proved that $\sum_{j=1}^{p} \lambda_j = \mathrm{tr}(\mathbf{X}^T\mathbf{X})$
- by definition we have $\phi_j^T \phi_j = 1$ and $\phi_i^T \phi_j = 0$ for $j \neq i$
- hence, λ_1 is the amount of the total variability represented by the first principal component.

The loading vector ϕ_1 (in other words, the eigenvector associated to the eigenvalue λ_1) defines a direction in feature space along which the data *vary* the most. Hence, the projection of the n points $\mathbf{Z}_1 = \mathbf{X}\phi_1$ (the first principal component) is formed by the principal component scores $z_{1,1}, z_{2,1}, \ldots, z_{n,1}$.

How many components? I

From the eigen-decomposition of $\mathbf{X}^T\mathbf{X}$ we can compute p eigenvalues and p eigenvectors. We sort the eigenvalues in descending order in such a way that the first principal component explains the most fraction of the variance $(\lambda_1 / \sum_{j=1}^{p} \lambda_j)$, the second principal component explains the second most fraction of the variance $(\lambda_2 / \sum_{j=1}^{p} \lambda_j)$ and so on.

There is not a formal rule to decide how many components take in consideration. In general, the user can chose among three strategies

How many components? II

- the screeplot
- bound on a given, a-priori chosen fraction of variability explained
- eigenvalue 1 (for standardized values: why?)

Points in $\mathcal{R}^n$ I

We see how represent the n points in the space of the variables $\mathcal{R}^p$. How represent variables in the space of individuals $\mathcal{R}^n$?

We have to maximize

$$\gamma_1^T \mathbf{X}\mathbf{X}^T \gamma_1 \text{ subject to } \gamma_1^T \gamma_1 = 1$$

obtaining $\mathbf{X}\mathbf{X}^T = \mu_1 \gamma_1$.

Here, μ_1 is the largest eigenvalue of the matrix $\mathbf{X}\mathbf{X}^T$, while γ_1 is the corresponding eigenvector.

Points in $\mathcal{R}^n$ II

It can be proved that $\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}^T\mathbf{X}$ have the same eigenvalues, so $\mathbf{X}\mathbf{X}^T = \lambda_1\gamma_1$.

It follows that the coordinates of the variables in the first principal component are

$$\mathbf{C}_1 = \mathbf{X}^T\gamma_1$$

We do not need to do eigendecomposition of $\mathbf{X}\mathbf{X}^T$. It can be proved that

$$\mathbf{X}^T\gamma_1 = \sqrt{\lambda_1}\phi_1$$

Eigendecomposition and Singular Value Decomposition I

The same analysis can be done through a singular value decomposition of the rectangular matrix $\mathbf{X}$

In a nutshell, we have that

$$\mathbf{X} = \mathbf{USV}^T,$$

where

- $\mathbf{U}$ is the matrix of the left singular vectors (corresponding to the $n \times n$ matrix γ. Usually, $n > p$, hence γ has dimension $n \times p$.)
- $\mathbf{S}$ is the diagonal matrix of the singular values (we have that $s_i = \sqrt{\lambda_i \times n}$ and $\lambda_i = s_i^2/n$, where s_i is the ith element of the matrix $\mathbf{S}$)
- $\mathbf{V}$ is the matrix of the right singular vectors (corresponding to the $p \times p$ matrix ϕ)

Eigendecomposition and Singular Value Decomposition II

It follows that:

- $\mathbf{Z} = \mathbf{US}$
- $\mathbf{C} = \mathbf{V}(\mathbf{S}/\sqrt{n})$

Let's compute a PCA by hand

PCA by hand I

```
> data(USArrests)
> #center the variables
> UAc <- as.matrix(scale(USArrests,scale=FALSE))
> n <- dim(UAc)[1]
> #covariance matrix
> covm <- crossprod(UAc/sqrt(n))
> #compute eigenvalues of covm
> eigd <- eigen(covm)
> #compute the first principal component
> z1 <- UAc %*% (matrix(eigd$vectors[,1],ncol=1))
> #what is the variance of the first principal component?
> sum(z1^2)/50
[1] 6870.893
> #the first eigenvalue is....
> eigd$values[1]
[1] 6870.893
```

PCA by hand II

```
    > #let's check with a R functions
    > require(FactoMineR)
    > sol1 <- PCA(USArrests,scale.unit=FALSE, graph=FALSE)
    > head(cbind(sol1$ind$coord[,1], z1 ) )
                [,1]       [,2]
Alabama     64.80216  -64.80216
Alaska      92.82745  -92.82745
Arizona    124.06822 -124.06822
Arkansas    18.34004  -18.34004
California 107.42295 -107.42295
Colorado    34.97599  -34.97599
    > #why there is a difference in sign?
```

PCA by hand III

```
    > #Now compute the projection of the variables on
    > #the first principal component
    > c1 <- sqrt(eigd$values[1])*eigd$vectors[,1]
    > #check with the result of PCA
    > cbind(sol1$var$coord[,1], c1 )
                              c1
Murder    3.456906  -3.456906
Assault  82.494735 -82.494735
UrbanPop  3.840809  -3.840809
Rape      6.229703  -6.229703
    > #compute all our stuffs
    > z <- UAc%*%eigd$vectors
    > c <- eigd$vectors%*%diag(sqrt(eigd$values))
```

PCA by hand IV

```
    > #Let's check
    > head(z)
                 [,1]       [,2]       [,3]       [,4]
Alabama     -64.80216  11.448007 -2.4949328  2.4079009
Alaska      -92.82745  17.982943 20.1265749 -4.0940470
Arizona    -124.06822  -8.830403 -1.6874484 -4.3536852
Arkansas    -18.34004  16.703911  0.2101894 -0.5209936
California -107.42295 -22.520070  6.7458730 -2.8118259
Colorado    -34.97599 -13.719584 12.2793628 -1.7214637
    > head(sol1$ind$coord)
               Dim.1      Dim.2      Dim.3      Dim.4
Alabama     64.80216 -11.448007 -2.4949328  2.4079009
Alaska      92.82745 -17.982943 20.1265749 -4.0940470
Arizona    124.06822   8.830403 -1.6874484 -4.3536852
Arkansas    18.34004 -16.703911  0.2101894 -0.5209936
California 107.42295  22.520070  6.7458730 -2.8118259
Colorado    34.97599  13.719584 12.2793628 -1.7214637
```

PCA by hand V

```
    > c
            [,1]        [,2]       [,3]        [,4]
[1,]  -3.456906   0.6306210  0.5132339  2.44535515
[2,] -82.494735   0.8267277 -0.4340818 -0.09570398
[3,]  -3.840809 -13.7439549 -1.2883503  0.14297025
[4,]  -6.229703  -2.8240149  6.2576925 -0.17776309
    > sol1$var$coord
             Dim.1      Dim.2      Dim.3       Dim.4
Murder    3.456906 -0.6306210  0.5132339  2.44535515
Assault  82.494735 -0.8267277 -0.4340818 -0.09570398
UrbanPop  3.840809 13.7439549 -1.2883503  0.14297025
Rape      6.229703  2.8240149  6.2576925 -0.17776309
```

PCA by hand VI

```
    > #now, proceed with the svd
    > sv <- svd(UAc)
    > names(sv)
[1] "d" "u" "v"
    > #d= singular values, u=left singuar vectors,
    > #v=right singular vectors
    > zz <- sv$u %*% diag(sv$d)
    > cc <- sv$v %*% diag(sqrt(1/n)*sv$d)
    > #we check again
```

PCA by hand VII

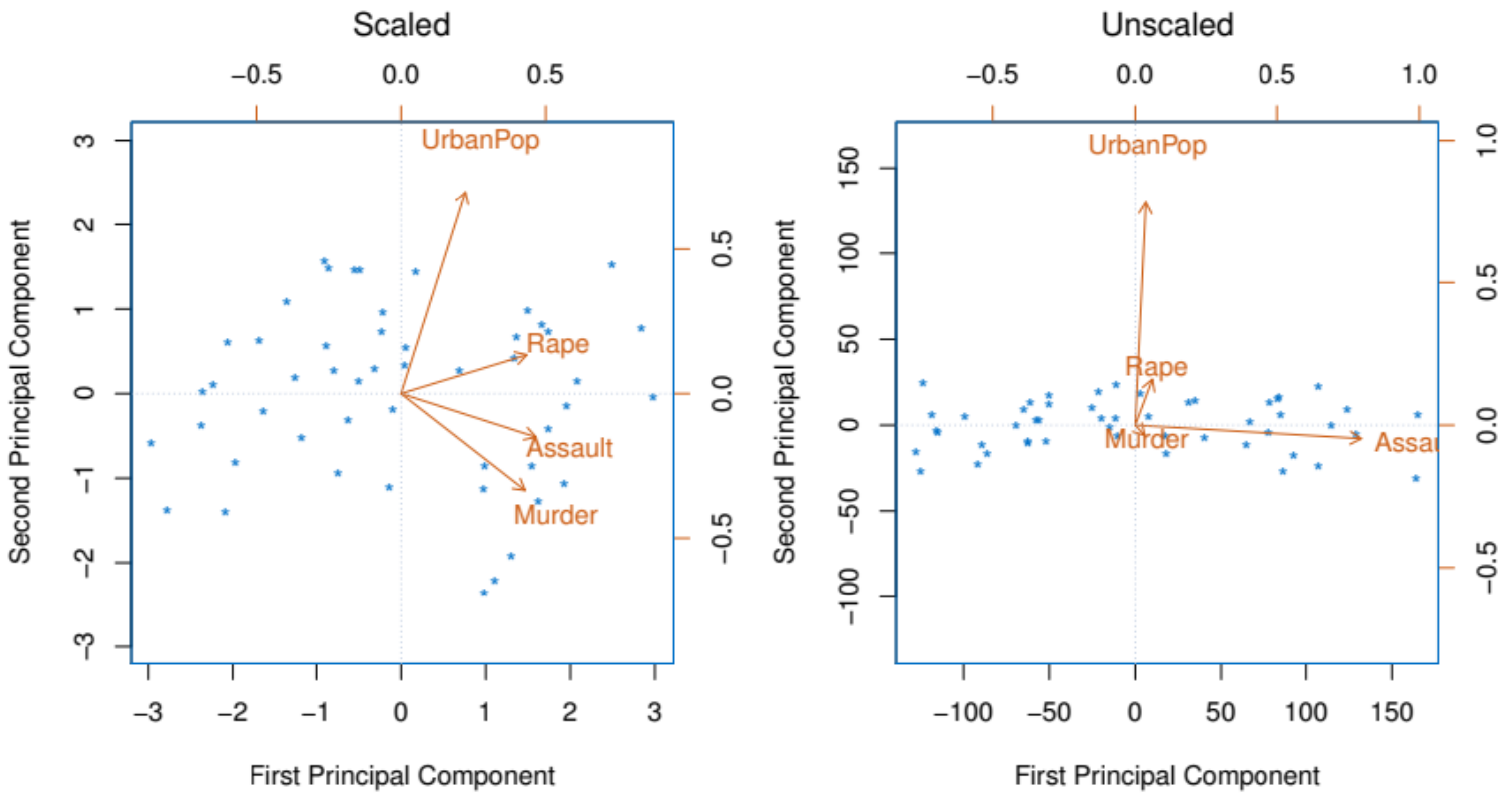
```
    > head(zz)
          [,1]       [,2]       [,3]       [,4]
[1,]  64.80216 -11.448007 -2.4949328 -2.4079009
[2,]  92.82745 -17.982943 20.1265749  4.0940470
[3,] 124.06822   8.830403 -1.6874484  4.3536852
[4,]  18.34004 -16.703911  0.2101894  0.5209936
[5,] 107.42295  22.520070  6.7458730  2.8118259
[6,]  34.97599  13.719584 12.2793628  1.7214637
    > head(sol1$ind$coord)
               Dim.1      Dim.2      Dim.3      Dim.4
Alabama     64.80216 -11.448007 -2.4949328  2.4079009
Alaska      92.82745 -17.982943 20.1265749 -4.0940470
Arizona    124.06822   8.830403 -1.6874484 -4.3536852
Arkansas    18.34004 -16.703911  0.2101894 -0.5209936
California 107.42295  22.520070  6.7458730 -2.8118259
Colorado    34.97599  13.719584 12.2793628 -1.7214637
```

PCA by hand VIII

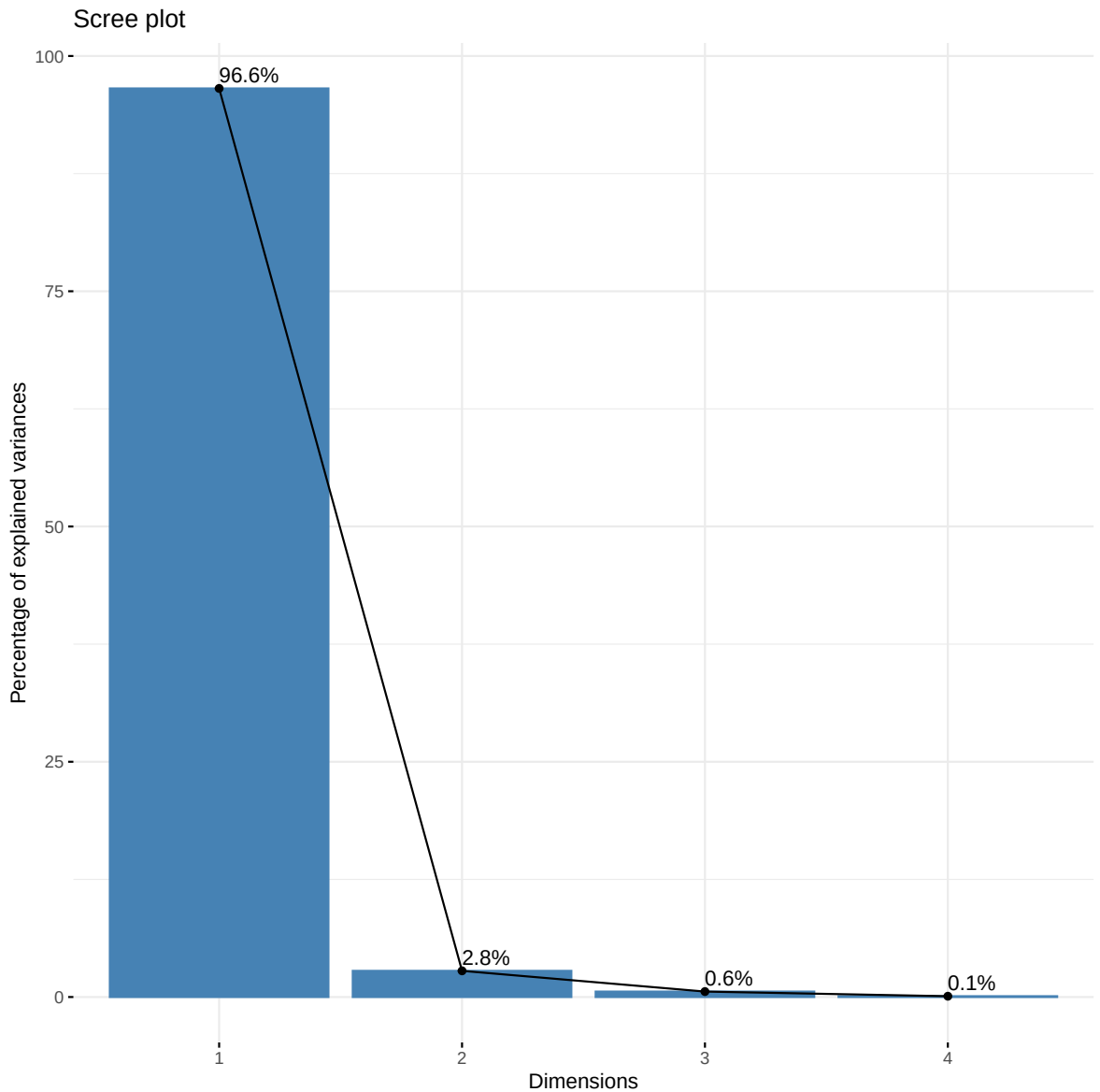
```
    > cc
           [,1]       [,2]       [,3]        [,4]
[1,]  3.456906 -0.6306210  0.5132339 -2.44535515
[2,] 82.494735 -0.8267277 -0.4340818  0.09570398
[3,]  3.840809 13.7439549 -1.2883503 -0.14297025
[4,]  6.229703  2.8240149  6.2576925  0.17776309
    > sol1$var$coord
             Dim.1      Dim.2      Dim.3       Dim.4
Murder    3.456906 -0.6306210  0.5132339  2.44535515
Assault  82.494735 -0.8267277 -0.4340818 -0.09570398
UrbanPop  3.840809 13.7439549 -1.2883503  0.14297025
Rape      6.229703  2.8240149  6.2576925 -0.17776309
```

Covariance matrix or correlation matrix?

- If the variables are in different units, scaling each to have standard deviation equal to one is recommended.
- If they are in the same units, you might or might not scale the variables.



Screeplot and variance explained



PCA and clustering

- Sometimes a cluster analysis is performed on a reduced data set after a PCA analysis. In fact, it is a clustering on the most important principal components. Don't forget that:
  1. PCA looks for a low-dimensional representation of the observations that explains a good fraction of the variance
  2. Clustering looks for homogeneous subgroups among the observations.