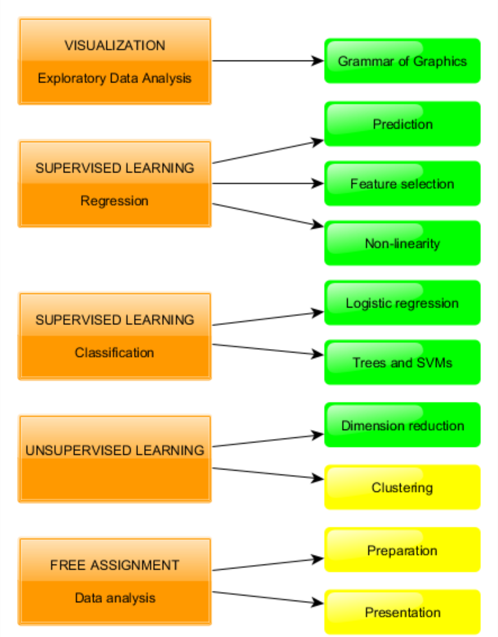
Unsupervised Learning

Principal Components Analysis

M. Cruyff

Program


VISUALIZATION
Exploratory Data Analysis
Grammar of Graphics
SUPERVISED LEARNING
Regression
Prediction
Feature selection
Non-linearity
SUPERVISED LEARNING
Classification
Logistic regression
Trees and SVMs
UNSUPERVISED LEARNING
Dimension reduction
Clustering
FREE ASSIGNMENT
Data analysis
Preparation
Presentation

Content

1. Unsupervised learning
2. Principal Components Analysis
3. Rotation

Unsupervised learning

Data

- no outcome variable Y
- only feature matrix **X**

Objectives

1. Dimension reduction
   - reveal data structure low-dimensional space
2. Clustering
   - find homogeneous subgroups

Principal Components Analysis (PCA)

What is PCA?

Principal Components Analysis

- transformation of p features $\rightarrow$ p principal components (PC's)

Principal Components

- directions in feature space
- maximize variance of feautures
- ordered in eigenvalues (variance explained)

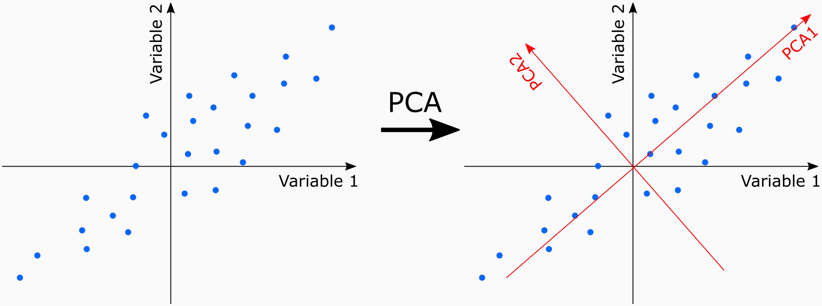
Component scores

- projections of data points onto PC's

Principal components

$$PC_j = \phi_{j1} X_1 + \phi_{jp} X_2, \quad j = 1, 2$$

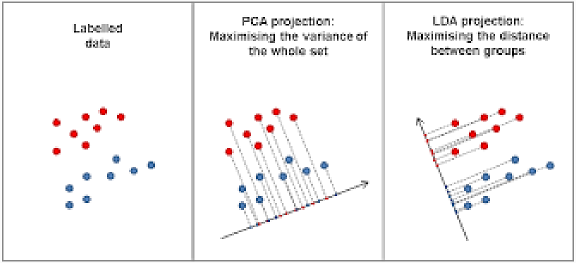
- ϕ_j are component loadings (correlation feature and PC)



Component scores

Projection onto principal component

- PCA maximizes variance between data points
- LDA maximizes variance between groups



Properties PC's

Based on covariance matrix features

- features with large variance dominate solution
  - weight in kilo versus length in cm
  - first PC dominated by length

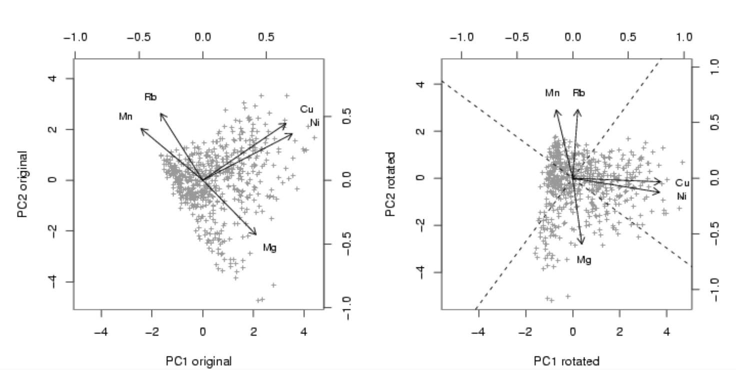
Based on correlation matrix features

- all features contribute equally
  - standardized scores of weight and length
  - both PC's combination of weight and length

Rotation of PC's

Facilitates interpretation of PC's

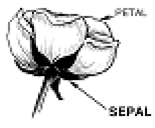
- maximizes loading on one PC
- minimizes loadings loadings on others



Example iris data

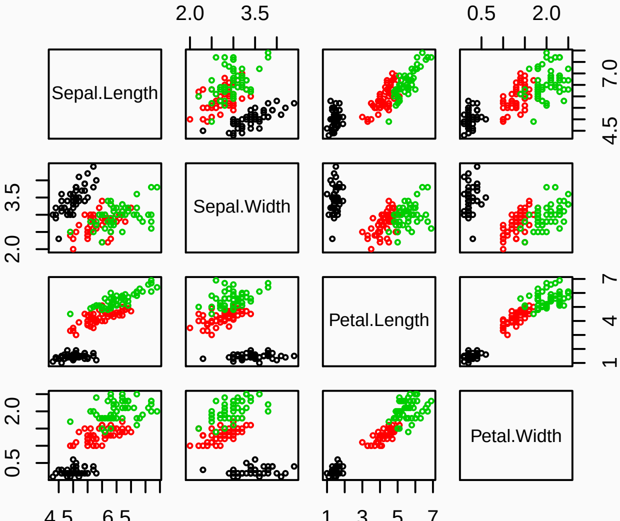
Iris flower data

- 150 Iris flowers (3 species)
- 4 features
  - length of petal and sepal
  - width of petal and sepal



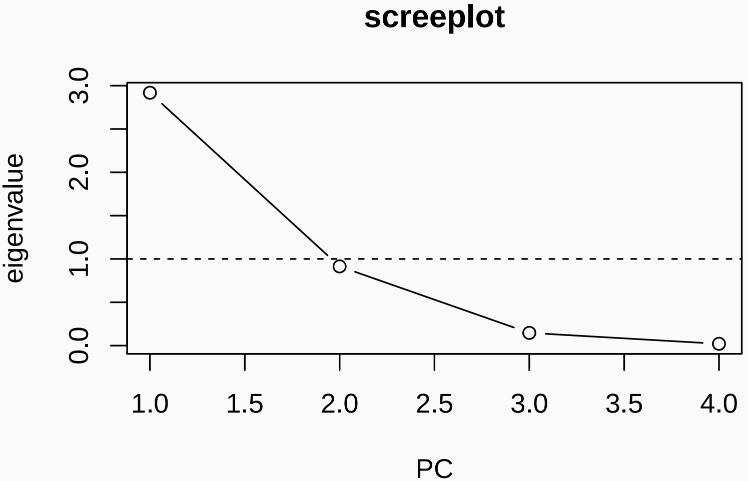
Scatter plot

How to summarize features structure in one or two dimensions?



How many PC's?

- Kaiser criterion: PC's with eigenvalue > 1
- Elbow criterion: PC's above elbow point



Component loadings

```
Loadings:
             PC1    PC2
Sepal.Length  0.890  0.361
Sepal.Width  -0.460  0.883
Petal.Length  0.992  0.023
Petal.Width   0.965  0.064

                 PC1   PC2
SS loadings    2.918 0.914
Proportion Var 0.730 0.229
Cumulative Var 0.730 0.958
```

PC1: dominated by sepal/petal length and petal width

PC2: dominated by sepal width

Rotated solutions

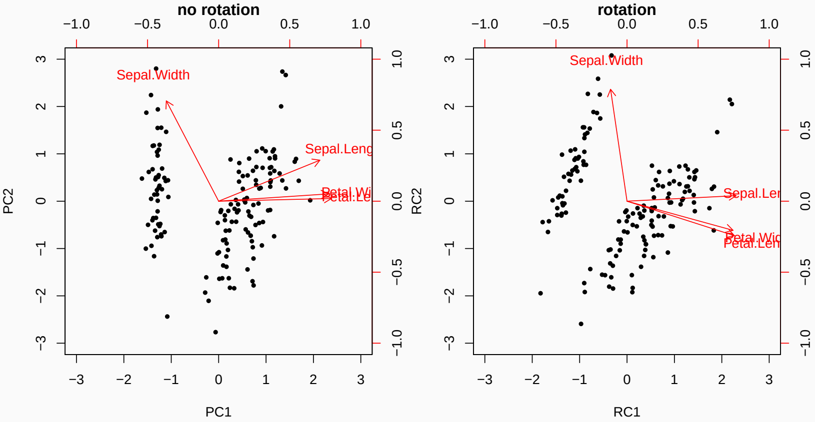
```
Loadings:
             RC1    RC2
Sepal.Length  0.959  0.048
Sepal.Width  -0.145  0.985
Petal.Length  0.944 -0.304
Petal.Width   0.932 -0.257

                 RC1   RC2
SS loadings    2.702 1.130
Proportion Var 0.676 0.283
Cumulative Var 0.676 0.958
```

PC's are more pronounced

- loading sepal width dropped from -0.460 to -0.145 on PC1
- loading sepal length dropped from 0.361 to 0.048 on PC2

Biplot

Loadings and component scores and in one plot



User-friendly with rotation

```
fit_pca <- principal(x        = <data>,
                     nfactors = <number>, # number to extract
                     rotate   = "none")   # default "varimax"

plot(fit_pca$values, type = "b")          # screeplot

print(loadings(fit_pca), cutoff = 0.4)    # abs(loadings) > 0.4

biplot(fit_pca, choose = <PC numbers>,    # sequence of 2 or more
       cutl = 0.4,                        # suppress labels for small loadings
       smoother = TRUE)                   # for large data sets
```

Handles data with $p > n$

```
fit_pca <- prcomp(x = <data>)

screeplot(fit_pca, type = "b")

print(loadings(fit_pca), cutoff = 0.4)    # abs(loadings) > 0.4

biplot(fit_pca)
```

Big data application PCA

Lab 4A

- Personality test bases on attitude towards professions
- Text analysis
- Face recognition