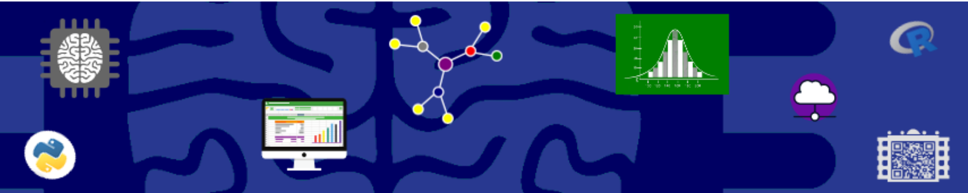
Introduction to Data Science

Winter Semester 2019/20

Oliver Ernst

TU Chemnitz, Fakultät für Mathematik, Professur Numerische Mathematik

Lecture Slides



Contents I

Contents II

Contents III

Contents

9 Unsupervised Learning

9.1 Principal Components Analysis

9.2 Clustering Methods

Unsupervised Learning

Introduction

- **Supervised learning**: n observations $\{(\boldsymbol{x}_i, y_i)_{i=1}^n\}$, each consisting of feature vector $\boldsymbol{x}_i \in \mathbb{R}^p$ and a response observation y_i.
- Construct prediction model $\hat{f}$ such that

$$y_i \approx \hat{f}(\boldsymbol{x}_i) \quad \text{in order to predict} \quad y = \hat{f}(\boldsymbol{x})$$

for values $\boldsymbol{x}$ not among data set.

Unsupervised Learning

Introduction

- **Supervised learning**: n observations $\{(\boldsymbol{x}_i, y_i)_{i=1}^n\}$, each consisting of feature vector $\boldsymbol{x}_i \in \mathbb{R}^p$ and a response observation y_i.
- Construct prediction model $\hat{f}$ such that

  $$y_i \approx \hat{f}(\boldsymbol{x}_i) \quad \text{in order to predict} \quad y = \hat{f}(\boldsymbol{x})$$

  for values $\boldsymbol{x}$ not among data set.
- **Unsupervised learning:** only feature observations available, no response data.
- Prediction not possible.

Unsupervised Learning

Introduction

- **Supervised learning**: n observations $\{(\boldsymbol{x}_i, y_i)_{i=1}^n\}$, each consisting of feature vector $\boldsymbol{x}_i \in \mathbb{R}^p$ and a response observation y_i.
- Construct prediction model $\hat{f}$ such that

$$y_i \approx \hat{f}(\boldsymbol{x}_i) \quad \text{in order to predict} \quad y = \hat{f}(\boldsymbol{x})$$

  for values $\boldsymbol{x}$ not among data set.
- **Unsupervised learning:** only feature observations available, no response data.
- Prediction not possible.
- Instead: statistical techniques for "discovering interesting things" about observations $\{\boldsymbol{x}_i\}_{i=1}^n$.
- Informative visualization of the data.
- Indentification of subgroups in the data/variables.
- Here: **principal components analysis** (PCA) and **clustering**.

Unsupervised Learning

Challenges

- For supervised learning tasks, e.g., binary classification, large selection of well developed algorithms (logistic regression, LDA, classification trees, SVMs) as well as assessment techniques (CV, validation set, . . .).

Unsupervised Learning

Challenges

- For supervised learning tasks, e.g., binary classification, large selection of well developed algorithms (logistic regression, LDA, classification trees, SVMs) as well as assessment techniques (CV, validation set, . . .).
- Unsupervised learning more subjective.
- No clear goal of analysis (such as response prediction).
- Often performed as part of **exploratory data analysis**.
- Results harder to assess (by very nature).
- Examples:
    - finding patterns in gene expression data for cancer patients;
    - identifying subgroups of customers of online shopping platform which display similar behavior/interest;
    - determining which content a search engine should display to which individuals.

Contents

9 Unsupervised Learning

9.1 Principal Components Analysis

9.2 Clustering Methods

- Many correlated feature/predictor variables $X_1, \dots, X_p$.
- Form new predictor variables Z_m (components) as linear combinations of original variables.
- Construct Z_m to be uncorrelated, ordered by decreasing variance.
- **Ideal situation:** first few $M < p$ components (principal components) explain large part of total variance of original variables. In this case data set well explained by restriction to principal components.

Unsupervised Learning

Principal components analysis

- Many correlated feature/predictor variables $X_1, \ldots, X_p$.
- Form new predictor variables Z_m (components) as linear combinations of original variables.
- Construct Z_m to be uncorrelated, ordered by decreasing variance.
- **Ideal situation:** first few $M < p$ components (principal components) explain large part of total variance of original variables. In this case data set well explained by restriction to principal components.
- Have used this idea for principal components regression (Chapter 6). There, used principal components as new (fewer) predictor variables.
- PCA: process by which principal components derived; also a technique for data visualization.
- Unsupervised, since applies only to feature/predictor variables.

Unsupervised Learning

Principal components

- To visualize p-variate data using bivariate scatterplots, $\binom{p}{2} = p(p-1)/2$ pairs to examine.
- Besides effort involved, individual scatterplots not necessarily that informative, containing only small fraction of information carried by complete data.
- Ideal: find low (1, 2 or 3)-dimensional representation of data containing all (most) relevant information.

- To visualize p-variate data using bivariate scatterplots, $\binom{p}{2} = p(p-1)/2$ pairs to examine.
- Besides effort involved, individual scatterplots not necessarily that informative, containing only small fraction of information carried by complete data.
- Ideal: find low (1, 2 or 3)-dimensional representation of data containing all (most) relevant information.
- **First principal component**: linear combination

$$Z_1 = \phi_{1,1}X_1 + \cdots + \phi_{p,1}X_p, \qquad \sum_{j=1}^{p} \phi_{j,1}^2 = 1, \tag{9.1}$$

  of original feature variables X_j with normalized coefficients ("**loadings**") with maximal variance.
  Loading vector $\boldsymbol{\phi}_1 := (\phi_{1,1}, \dots, \phi_{p,1})^\top$.

Computing the first principal component

- Given data set

$$\boldsymbol{X} \in \mathbb{R}^{n \times p}, \quad \text{i.e., } n \text{ samples of } p \text{ features } X_1, \dots, X_p,$$

- Each column $\boldsymbol{x}_j = (x_{1,j}, \dots, x_{n,j})^\top \in \mathbb{R}^n$, $j = 1, \dots, p$, contains n samples (observations) of j-th feature.
- Each row $\tilde{\boldsymbol{x}}_i^\top = (x_{i,1}, \dots, x_{i,p}) \in \mathbb{R}^p$, $i = 1, \dots, n$, contains one sample of p features.
- Here information synonymous with variance, hence assume centered columns, i.e.,

$$\boldsymbol{e}^\top \boldsymbol{x}_j = 0, \quad j = 1, \dots, p, \qquad \boldsymbol{e} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^n,$$

hence sample mean of each column is zero.

Computing the first principal component

- Loadings $\{\phi_{j,1}\}_{j=1}^{p}$ for first principal component determined as (normalized) coefficients in linear combination

$$z_1 = \phi_{1,1}x_1 + \cdots + \phi_{p,1}x_p = \boldsymbol{X}\boldsymbol{\phi}_1$$

such that z_1 has largest sample variance (mean remains zero).

Unsupervised Learning

Computing the first principal component

- Loadings $\{\phi_{j,1}\}_{j=1}^p$ for first principal component determined as (normalized) coefficients in linear combination

$$z_1 = \phi_{1,1}\boldsymbol{x}_1 + \cdots + \phi_{p,1}\boldsymbol{x}_p = \boldsymbol{X\phi}_1$$

  such that $\boldsymbol{z}_1$ has largest sample variance (mean remains zero).

- In other words, loadings $\{\phi_{j,1}\}_{j=1}^p$ solve optimization problem

$$\max \left\{ \frac{1}{n} \sum_{i=1}^{n} \left(\sum_{j=1}^{p} \phi_{j,1} x_{i,j} \right)^2 : \sum_{j=1}^{p} \phi_{j,1}^2 = 1 \right\} \tag{9.2}$$

Unsupervised Learning

Computing the first principal component

- Loadings $\{\phi_{j,1}\}_{j=1}^{p}$ for first principal component determined as (normalized) coefficients in linear combination
$$z_1 = \phi_{1,1}\boldsymbol{x}_1 + \cdots + \phi_{p,1}\boldsymbol{x}_p = \boldsymbol{X}\boldsymbol{\phi}_1$$
such that $\boldsymbol{z}_1$ has largest sample variance (mean remains zero).
- In other words, loadings $\{\phi_{j,1}\}_{j=1}^{p}$ solve optimization problem
$$\max\left\{\frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j,1}x_{i,j}\right)^{2} : \sum_{j=1}^{p}\phi_{j,1}^{2} = 1\right\} \tag{9.2}$$
- In other words, loading vector $\boldsymbol{\phi}_1$ solves optimization problem
$$\max_{\|\boldsymbol{\phi}\|_2=1}\|\boldsymbol{X}\boldsymbol{\phi}\|_2^2 = \max_{\|\boldsymbol{\phi}\|_2=1}\boldsymbol{\phi}^\top\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{\phi}.$$
- In other words (Courant-Fischer max-min principle), $\boldsymbol{\phi}_1$ is a normalized eigenvector associated with largest eigenvalue of $\boldsymbol{X}^\top\boldsymbol{X}$.

Unsupervised Learning

Computing the first principal component

- Equivalent characterization: $\boldsymbol{\phi}_1$ is a right singular vector associated with the largest singular values of (centered) data matrix $\boldsymbol{X}$.

Unsupervised Learning

Computing the first principal component

- Equivalent characterization: $\boldsymbol{\phi}_1$ is a right singular vector associated with the largest singular values of (centered) data matrix $\boldsymbol{X}$.
- Components $z_{1,1}, \dots, z_{n,1}$ of $\boldsymbol{z}_1$ referred to as **scores** of first principal component.

Unsupervised Learning

Computing the first principal component

- Equivalent characterization: $\boldsymbol{\phi}_1$ is a right singular vector associated with the largest singular values of (centered) data matrix $\boldsymbol{X}$.
- Components $z_{1,1}, \dots, z_{n,1}$ of $\boldsymbol{z}_1$ referred to as **scores** of first principal component.
- **Geometric interpretation:** loading vector $\boldsymbol{\phi}_1$ defines direction in feature space along which data varies the most.

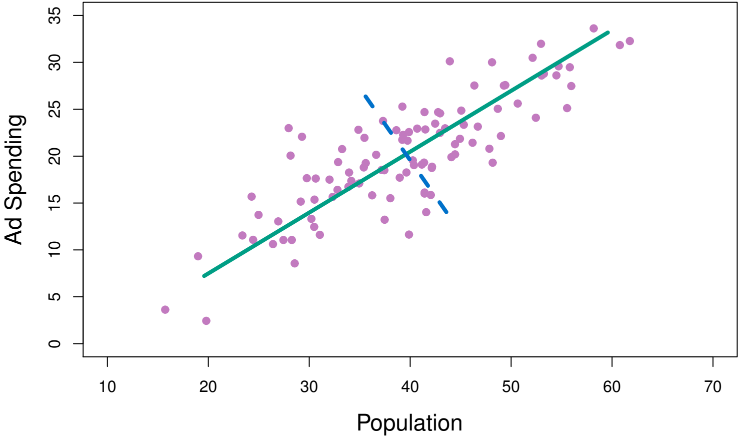
  "Projection of data points $\tilde{\boldsymbol{x}}_1, \dots, \tilde{\boldsymbol{x}}_n$ (rows of $\boldsymbol{X}$) in this direction yield principal component scores $\boldsymbol{z}_1$."

  This is simply the dual interpretation of the matrix-vector product $\boldsymbol{z}_1 = \boldsymbol{X}\boldsymbol{\phi}_1$: rather than as a linear combination of the columns $\{\boldsymbol{x}_j\}_{j=1}^p \subset \mathbb{R}^n$ of $\boldsymbol{X}$, it is viewed as the vector of inner products of $\boldsymbol{\phi}_1$ with the rows $\{\tilde{\boldsymbol{x}}_i\}_{i=1}^n \subset \mathbb{R}^{1\times p}$ of $\boldsymbol{X}$:

$$\boldsymbol{z}_1 = \boldsymbol{X}\boldsymbol{\phi}_1 = \begin{bmatrix} \tilde{\boldsymbol{x}}_1^\top \boldsymbol{\phi}_1 \\ \vdots \\ \tilde{\boldsymbol{x}}_n^\top \boldsymbol{\phi}_1 \end{bmatrix}.$$

Unsupervised Learning

Computing the first principal component



First principal component loading vector in advertising data set (green). Here $p = 2$ and observation data can be viewed along with principal component vectors.

Unsupervised Learning

Computing the second principal component

- Second principal component Z_2: linear combination of $X_1, \dots, X_p$ with largest variance subject to condition that it is uncorrelated with Z_1.
- Scores

$$z_2 = \phi_{1,2} x_1 + \dots + \phi_{p,2} x_p = \boldsymbol{X}\boldsymbol{\phi}_2$$

  with second principal components loading vector $\boldsymbol{\phi}_2 = (\phi_{1,2}, \dots, \phi_{p,2})^\top$.
- Uncorrelatedness equivalent with orthogonality in Euclidean inner product.
- Hence $\boldsymbol{\phi}_2$ is normalized eigenvector associated with second-largest eingenvalue of $\boldsymbol{X}^\top \boldsymbol{X}$, or normalized right singular vector associated with second-largest singular value of $\boldsymbol{X}$.
- Previous figure: $p = 2$, only one possibility for $\boldsymbol{\phi}_2$ (dashed blue line).
- Remaining components Z_m defined analogously: linear combination of $X_1, \dots, X_p$ with maximal variance uncorrelated with $Z_1, \dots, Z_{m-1}$ (Euclidean orthogonality of recombined sample vectors).
- There are at most $\min\{n-1, p\}$ principal components.

- Denoting the SVD of the centered data matrix as $\boldsymbol{X} = \boldsymbol{U\Sigma V}^\top$ gives $\boldsymbol{X}^\top\boldsymbol{X} = \boldsymbol{V\Sigma}^\top\boldsymbol{\Sigma V}^\top$.
- The eigenvalues of $\boldsymbol{X}^\top\boldsymbol{X}$ in descending order are displayed on the diagonal of $\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1^2, \dots, \sigma_p^2)$.
- The total variance in the data represented by $\boldsymbol{X}$ is given by $\|\boldsymbol{X}\|_F^2 = \sigma_1^2 + \dots + \sigma_p^2$.
- The principal component loading vectors $\{\boldsymbol{\phi}_j\}_{j=1}^p$ are given by the normalized eigenvectors of $\boldsymbol{X}^\top\boldsymbol{X}$ or, equivalently, the right singular vectors of $\boldsymbol{X}$, i.e.,

$$\boldsymbol{\phi}_j = \boldsymbol{v}_j, \qquad j = 1, \dots, \min\{n-1, p\}.$$

- For the scores $\boldsymbol{z}_m$, we have

$$\boldsymbol{z}_m = \boldsymbol{X}\boldsymbol{\phi}_m = \boldsymbol{U\Sigma V}^\top\boldsymbol{v}_m = \sigma_m\boldsymbol{u}_m, \qquad m = 1, \dots, \min\{n-1, p\}.$$

Unsupervised Learning

Example: USArrests data set

- USArrests data set: arrests per 100,000 residents of each of the 50 states of the USA for each of the crimes Assault, Murder and Rape.
- Also records UrbanPop, percentage of each state's population living in urban areas.

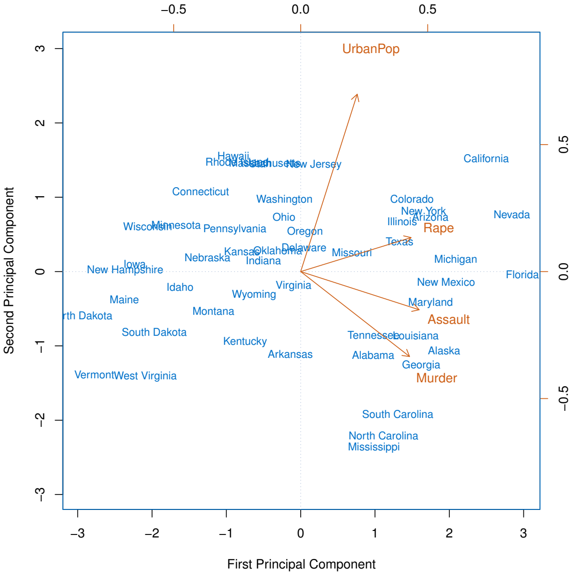
Example: USArrests data set

- USArrests data set: arrests per 100,000 residents of each of the 50 states of the USA for each of the crimes Assault, Murder and Rape.
- Also records UrbanPop, percentage of each state's population living in urban areas.
- Number of samples = length of PC score vector $n = 50$.
- Dimension of feature space = length of PC loading vectors $p = 4$.
- PCA performed after standardizing data matrix (column mean zero, standard deviation one).
- PC loading vectors

	PC1	PC2
Murder	0.5358995	-0.4181809
Assault	0.5831836	-0.1879856
Rape	0.5434321	0.1673186
UrbanPop	0.2781909	0.8728062

Unsupervised Learning

Example: `USArrests` data set



- **Biplot** of data in space of first two principal components.
- Blue state names: score in first 2 PC.
- Orange arrows: first two PC loading vectors (axes on right and top).
- Biplot: displays both PC scores and PC loadings.

Unsupervised Learning

Example: USArrests data set: Interpretation of figure

- First loading vector places approximately equal weight on Assault, Murder and Rape, much less weight on UrbanPop.
  Hence first PC roughly corresponds to measure of overall rate of serious violent crime.

Unsupervised Learning

Example: USArrests data set: Interpretation of figure

- First loading vector places approximately equal weight on `Assault`, `Murder` and `Rape`, much less weight on `UrbanPop`.
  Hence first PC roughly corresponds to measure of overall rate of serious violent crime.
- Second loading vector has more weight on `UrbanPop`, much less on remaining three features, hence roughly corresponds to level of urbanization of each state.

Unsupervised Learning

Example: USArrests data set: Interpretation of figure

- First loading vector places approximately equal weight on Assault, Murder and Rape, much less weight on UrbanPop.
  Hence first PC roughly corresponds to measure of overall rate of serious violent crime.
- Second loading vector has more weight on UrbanPop, much less on remaining three features, hence roughly corresponds to level of urbanization of each state.
- Overall, crime-related variables close to each other in space spanned by first two PC, UrbanPop far from these: indicates crime-related variables highly correlated, weakly correlated with UrbanPop.

Unsupervised Learning

Example: USArrests data set: Interpretation of figure

- First loading vector places approximately equal weight on Assault, Murder and Rape, much less weight on UrbanPop.
  Hence first PC roughly corresponds to measure of overall rate of serious violent crime.
- Second loading vector has more weight on UrbanPop, much less on remaining three features, hence roughly corresponds to level of urbanization of each state.
- Overall, crime-related variables close to each other in space spanned by first two PC, UrbanPop far from these: indicates crime-related variables highly correlated, weakly correlated with UrbanPop.
- State differences in first PC: states with high score in first component tend to have high crime rates (e.g. California, Nevada, Florida); those with negative first PC scores tend to have low crime rates (e.g. North Dakota).
- State differences in 2nd PC: High score in 2nd PC (e.g. California) indicates high level uf urbanisation, low score low level (e.g. Mississippi).

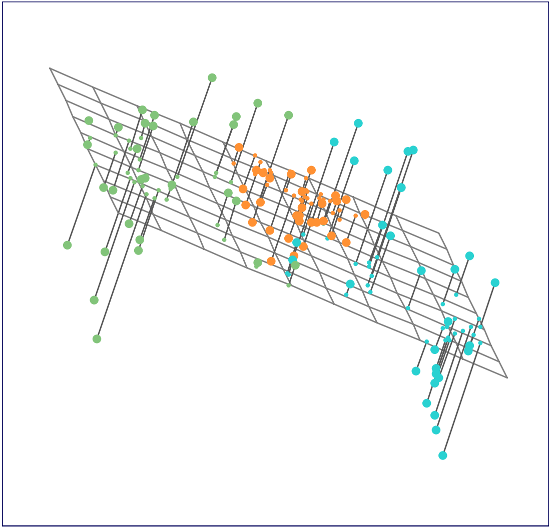
Unsupervised Learning

Example: USArrests data set: Interpretation of figure

- First loading vector places approximately equal weight on Assault, Murder and Rape, much less weight on UrbanPop.
  Hence first PC roughly corresponds to measure of overall rate of serious violent crime.
- Second loading vector has more weight on UrbanPop, much less on remaining three features, hence roughly corresponds to level of urbanization of each state.
- Overall, crime-related variables close to each other in space spanned by first two PC, UrbanPop far from these: indicates crime-related variables highly correlated, weakly correlated with UrbanPop.
- State differences in first PC: states with high score in first component tend to have high crime rates (e.g. California, Nevada, Florida); those with negative first PC scores tend to have low crime rates (e.g. North Dakota).
- State differences in 2nd PC: High score in 2nd PC (e.g. California) indicates high level uf urbanisation, low score low level (e.g. Mississippi).
- States close to origin?

Unsupervised Learning

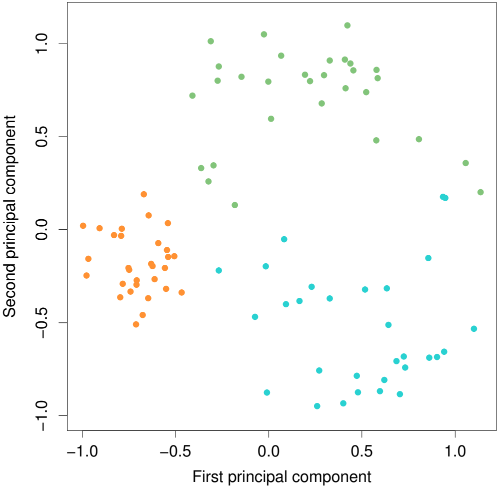
PCA: another interpretation



- First two PC loading vectors of a 3D data set along with observations.
- Span a plane along which observations have highest variance.
- Alternative interpretation: PC provide low-dimensional surfaces that are **closest** to the observations.

Unsupervised Learning

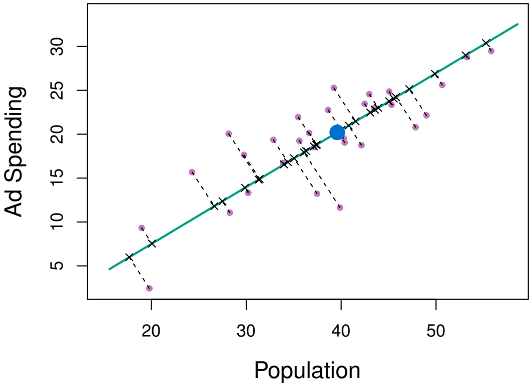
PCA: another interpretation



- First two PC loading vectors of a 3D data set along with observations.
- Span a plane along which observations have highest variance.
- Alternative interpretation: PC provide low-dimensional surfaces that are **closest** to the observations.
- Projection of observations to closest plane: variance is maximized.

Unsupervised Learning

PCA: another interpretation



- Example from Chapter 6 (ad spending vs. population)
- 1st PC loading vector: line in $\mathbb{R}^p$ closest to observations (in Euclidean distance).
- Dashed lines: distance between each observation and first PC loading vector.
- In this sense: good summary of the data.

Unsupervised Learning

PCA: another interpretation

Summary: the first M principal components and associated score vectors together yield a best approximation of the observational data:

$$x_{i,j} \approx \sum_{m=1}^{M} z_{i,m}\, \phi_{j,m}. \tag{9.3}$$

Summary: the first M principal components and associated score vectors together yield a best approximation of the observational data:

$$x_{i,j} \approx \sum_{m=1}^{M} z_{i,m}\, \phi_{j,m}. \tag{9.3}$$

Explanation: writing all $n \times p$ equations (9.3) in matrix form yields

$$\boldsymbol{X} \approx \sum_{m=1}^{M} \boldsymbol{z}_m \boldsymbol{\phi}_m^\top = \sum_{m=1}^{M} \sigma_m \boldsymbol{u}_m \boldsymbol{v}_m^\top,$$

which is simply the singular value expansion of $\boldsymbol{X}$ truncated after m terms. Recalling the best approximation property of the truncated SVD in the spectral and Frobenius norms explains the nearness of the expression (9.3) to the data.

Unsupervised Learning

PCA: scaling

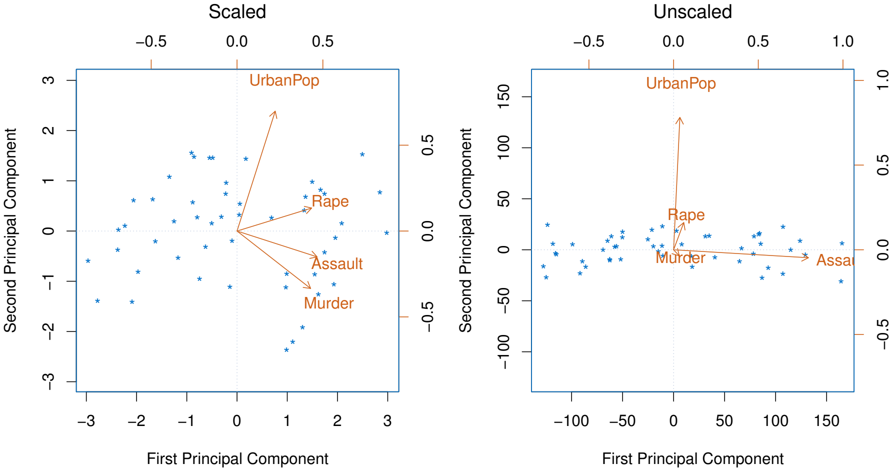
- Data matrix centered before applying PCA.
- Individual scaling of the predictor variables (columns) will affect the outcome of PCA.
- Contrast with linear regression, where rescaling of a variable exactly compensated by associated coefficient.
- In `USArrests` example, each variable was rescaled to have standard deviation one.
- Reason: variables have different units (`Murder`, `Rape`, and `Assault` in # occurrences / 100,000 people, `UrbanPop` in percentage living in urban areas.

  Also: variances 18.97, 87.73, 6945.16 and 209.5, respectively, display large variation.

  Hence without scaling, first PC loading vector will have very large weight on `Assault`.
- Scaling is recommended, but doing so should be deliberate.

Unsupervised Learning

PCA: scaling



Biplots for PCA applied to `USArrests` data set.

Left: all variables scaled to have standard deviation one.

Right: PCA performed on unscaled variables.

Unsupervised Learning

PCA: uniqueness

Unsupervised Learning

PCA: uniqueness

- Singular vectors, normalized eigenvectors unique up to sign.
  Hence same holds for principal components.
- Different software packages will yield same PC loading vectors up to sign.
- Sign flipping harmless, as PC represent directions in Euclidean space.
- Note that flipping sign in $\boldsymbol{\phi}_m$ in (9.3) will result in sign flip in $\boldsymbol{z}_m$, leaving product unchanged.

Unsupervised Learning

PCA: proportion of variance explained

- How much information is lost by replacing original data with PC approximation (projecting observations on first $M < p$ principal componants)?

Unsupervised Learning

PCA: proportion of variance explained

- How much information is lost by replacing original data with PC approximation (projecting observations on first $M < p$ principal componants)?
- More precisely: how much of the variance of the original data is missing in the PC approximation? What is the **portion of variance explained** (PVE)?
- Define total variance in (centered) $\boldsymbol{X}$ by

$$\sum_{j=1}^{p} \mathbf{Var}\, X_j := \sum_{j=1}^{n} \frac{1}{n} \sum_{i=1}^{n} x_{i,j}^2 = \frac{1}{n} \|\boldsymbol{X}\|_F^2.$$

- Variance explained by m-th PC:

$$\frac{1}{n}\|\boldsymbol{z}_m\|_2^2 = \frac{1}{n}\sum_{i=1}^{n} z_{i,m}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j,m}x_{i,j}\right)^2 = \|\boldsymbol{X}\boldsymbol{\phi}_m\|_2^2.$$

- Hence PVE of m-th PC given by $\|\boldsymbol{X}\boldsymbol{\phi}_m\|_2^2 / \|\boldsymbol{X}\|_F^2$.

Unsupervised Learning

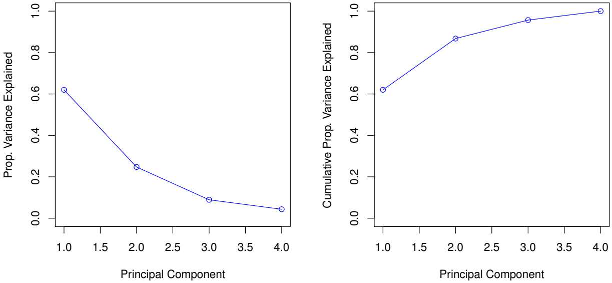
PCA: proportion of variance explained

- In `USArrests` data set: first PC explains 62% of total variance, 2nd explains 24.7%. Hence first two explain ≈ 87%, remaining two only 13%.
- Therefore, the biplot gives an accurate summary of the data (using just 2 dimensions).

Unsupervised Learning

PCA: proportion of variance explained

- In USArrests data set: first PC explains 62% of total variance, 2nd explains 24.7%. Hence first two explain ≈ 87%, remaining two only 13%.
- Therefore, the biplot gives an accurate summary of the data (using just 2 dimensions).
- **Scree plots**: display PVE of each PC as well as cumulative PVE



Unsupervised Learning

PCA: sufficient number of principal components

- Can choose M between 1 and $\min\{p, n-1\}$.

Unsupervised Learning

PCA: sufficient number of principal components

- Can choose M between 1 and $\min\{p, n-1\}$.
- Ideal: smallest M conveying good understanding of data.
- Scree plot can provide guidance: fix M at "**elbows**", i.e., where proportion of variance explained has a noticeable drop.
  In previous example, elbow after $M = 2$ could be argued.
- Such visual analysis is heuristic, subjective and ad-hoc, but there is no general answer for determining how many PCs is enough (exploratory data analysis).
- In supervised learning, M is a tuning parameter, which can be determined by CV or similar validation technique.

Unsupervised Learning

PCA: further uses for PC

- Supervised learning: new features, smaller in number than original.
- **Low-rank** approximation of $\boldsymbol{X}$ obtained by truncating SVD after $M < p$ terms often better than full $\boldsymbol{X}$ due to noise reduction (e.g. latent semantic indexing).
- Signal of a data set often contained in first few PC, rest can be noise.

Contents

9 Unsupervised Learning

9.1 Principal Components Analysis

9.2 Clustering Methods

Unsupervised Learning

Clustering methods

- Broad set of techniques for finding **clusters** or **subgroups** in a data set.
- Partition data into distinct subsets of similar observations, where notion of similarity is problem-dependent.
- Unsupervised problem of finding structure in data set.
- Clustering and PCA seek to simplify data via small number of summaries, but via different mechanisms
  - PCA seeks low-dimensional representation of observations explaining good fraction of their variance.
  - Clustering seeks homogeneous subgroups among observations.
- **Example:** given marketing measurements (median household income, occupation, distance from nearest urban area, etc.) for large population, perform *market segmentation* to identify subgroups of people more receptive to a particular form of advertising or more likely to buy a particular product (cluster people in a data set)
- Here: 2 approaches, **K-means clustering**, **hierarchical clustering**.

Unsupervised Learning

K-means clustering

- Partition data into $K \in \mathbb{N}$ disjoint clusters.
- Upon fixing K, algorithm assigns each observation to one of K clusters.

Unsupervised Learning

K-means clustering

- Partition data into $K \in \mathbb{N}$ disjoint clusters.
- Upon fixing K, algorithm assigns each observation to one of K clusters.
- Let $\{C_k\}_{k=1}^K$ denote sets containing indices of n observations in cluster k such that

$$\begin{aligned} &\bigcup_{k=1}^{K} C_k = \{1, \dots, n\}, \\ &C_k \cap C_\ell = \emptyset \text{ for } k \neq \ell, \qquad k, \ell = 1, \dots, K, \end{aligned}$$

- A good clustering is one for which *within-cluster variation* is small.
- With $W(C_k)$ denoting a measure of amount by which observations in cluster k differ, K-means clustering tries to determine

$$\operatorname*{arg\,min}_{C_1,\dots,C_K} \sum_{k=1}^{K} W(C_k).$$

Unsupervised Learning

K-means clustering

- Common measure for in-cluster-variation: **squared Euclidean distance**

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{i,j} - x_{i',j})^2,$$

  $|C_k|$ denoting the cardinality of C_k.

- The cluster optimization problem thus becomes

$$\operatorname*{arg\,min}_{C_1,\dots,C_K} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{i,j} - x_{i',j})^2 \right\} \tag{9.4}$$

- The number of possible clusterings of n observations in to K clusters grows [10] like K^n. There are, however, simple heuristics for finding good approximations of the solution.

[10] These are known as the *Stirling numbers of the second kind*, $S(n,K) \sim k^n/k!$ as $n \to \infty$.

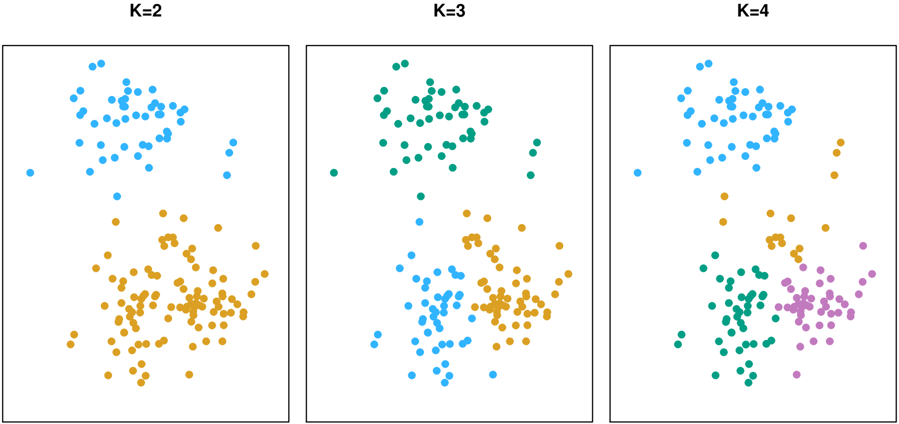
Algorithm 6: *K*-means clustering.

```
1 Randomly assign a number, from 1 to $K$, to each of the observations.
  These serve as initial cluster assignments for the observations.
2 Iterate until the cluster assignments stop changing:
    a For each of the $K$ clusters, compute the cluster centroid. The $k$-th cluster
      centroid is the vector of the $p$ feature means for the observations in the $k$th cluster.
    b Assign each observation to the cluster whose centroid is closest (where closest
      is defined by Euclidean distance).
```

The name of the algorithm is due to the computation of the centroids in step (2a), which are computed as the mean across all observations currently assigned to each cluster.

Unsupervised Learning

K-means clustering



Simulated data in 2D, $n = 150$. Results of applying K-means clustering with $K = 2, 3, 4$.

- Algorithm 6 guaranteed to decrease the value of the objective (9.4) in each step.
- Introducing the **cluster means**

$$\overline{x}_{k,j} := \frac{1}{|C_k|} \sum_{i \in C_k} x_{i,j}, \qquad j = 1, \dots, p,$$

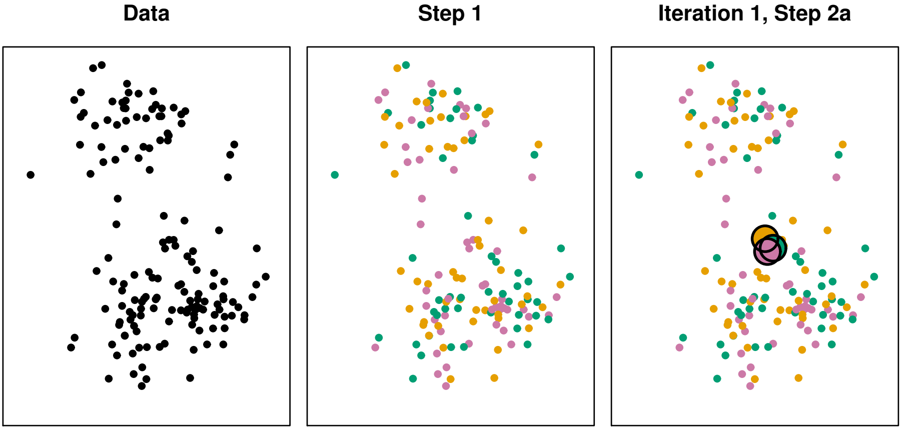
  there holds

$$\frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{i,j} - x_{i',j})^2 = 2 \sum_{i \in C_k} \sum_{j=1}^{p} (x_{i,j} - \overline{x}_{k,j})^2.$$

- In Step (2a), cluster means for each feature are the constants that minimizing the sum-of-squares deviations.
- In step (2b), reallocating the observations within the clusters can only decrease the objective.
- As Algorithm 6 is run, objective improves until it no longer changes, ending in a **local optimum**.

Unsupervised Learning

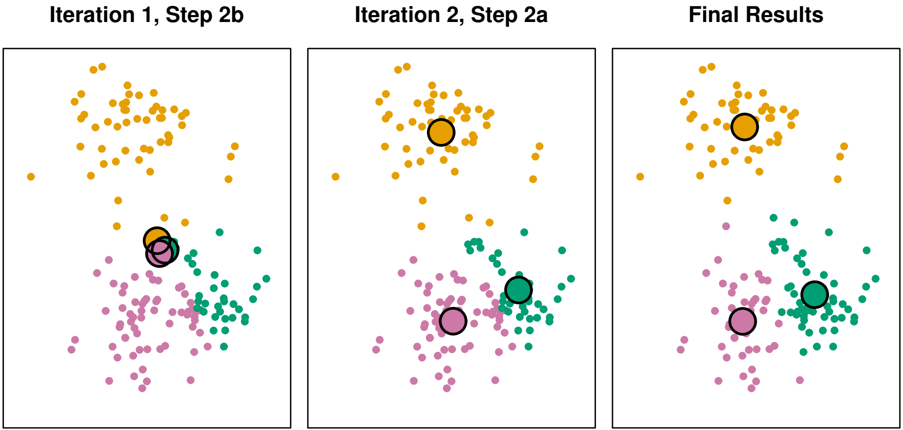
K-means clustering



Progress of K-means algorithm for running example, $K = 3$, beginning with just observations, initial random assignment to clusters, centroid computation (large colored disks), reassignment to clusters, recomputation of centroid, and final result after 10 iterations.

Unsupervised Learning

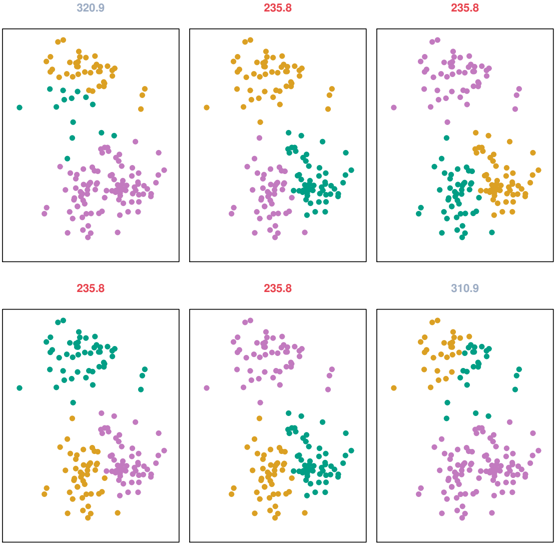
K-means clustering



Progress of K-means algorithm for running example, $K = 3$, beginning with just observations, initial random assignment to clusters, centroid computation (large colored disks), reassignment to clusters, recomputation of centroid, and final result after 10 iterations.

Unsupervised Learning

K-means clustering, dealing with local minima



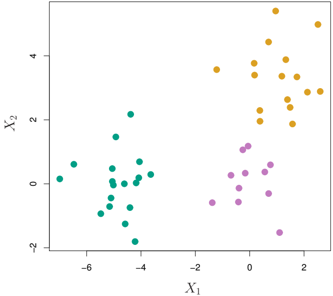
Since result of K-means typically only local minimum, advisable to run multiple times using different random initial clusterings and pick the outcome with smallest objective.

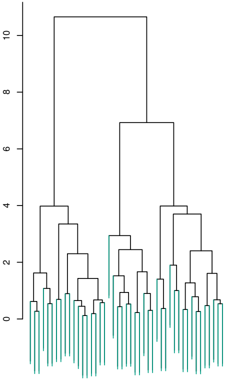
Here K-means with $K = 3$ was run on the data in the previous toy example with different random initializations. Three outcomes achieved the same (suboptimal) objective value.

Unsupervised Learning

Hierarchical clustering

- Alternative to K-means algorithm, does not require K to be specified in advance.
- Results in tree-based cluster representation called a **dendrogram**.
- Here: **bottom-up** or **agglomerative clustering**.





Simulated data, 45 observations, 3 classes, hierarchical clustering results in dendrogram on the right.

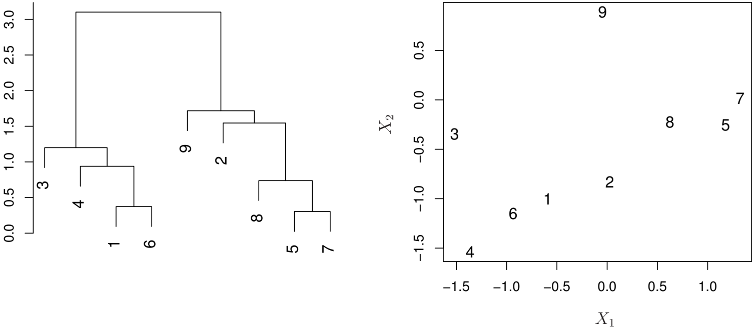
Unsupervised Learning

Hierarchical clustering, interpreting a dendrogram

- Each leaf corresponds to one of the original 45 observations.
- Moving up the tree, some leaves begin to fuse into branches, reflecting similarity of the leaves.
- Advancing further up, branches fuse with leaves or other branches.
- Earlier fusion (bottom-up) indicates stronger similarity of (groups of) observations.
- More precisely: for any pair of observations, the distance (from bottom) to where their subtrees are first joined is a measure of their non-similarity.

Unsupervised Learning

Hierarchical clustering, interpreting a dendrogram

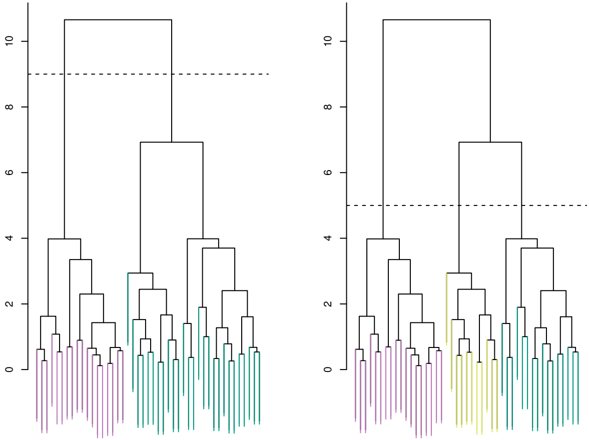


Left: dendrogram of 9 observations of two-dimensional data. Right: Original data. 1 and 6 as well as 5 and 7 very similar; 9 no more similar to 2 than to 8, 5 and 7, even though 9 and 2 close horizontally in dendrogram; 2, 8, 5, 7 all fuse with 9 at same height, ≈ 1.8.

Unsupervised Learning

Hierarchical clustering, identifying clusters from a dendrogram

Cutting a dendrogram horizontally, the distinct sets of observations beneath the cut can be interpreted as clusters.



On the left, cutting dendrogram at height of 9 yields 2 clusters.

On the right, cutting at height 5 yields 3 clusters.

Further cuts can be made at different heights yielding clusters of size between 1 (no cut) and n (cut at height 0).

Height plays same role as K in K-means clustering.

Unsupervised Learning

Hierarchical clustering, identifying clusters from a dendrogram

- Single dendrogram yields any number of clusterings.
- Cut usually chosen by inspection.
- **Hierarchical** refers to the fact that clusters from different heights in the same dendrogram are nested. However, nested structure not always realistic. (Group split 50-50 among males and females, and equally split among 3 nationalities.) In such situations K-means may yield better results.

Unsupervised Learning

Hierarchical clustering algorithm

- Introduce measure of dissimilarity between observation pairs. e.g. Euclidean distance.
- Start at bottom, each observation treated as its own cluster.
- Two most similar clusters fused, yielding $n - 1$ clusters.
- Next fusion yields $n - 2$ clusters.
- Proceed until single cluster remains.

Unsupervised Learning

Hierarchical clustering algorithm

- Introduce measure of dissimilarity between observation pairs. e.g. Euclidean distance.
- Start at bottom, each observation treated as its own cluster.
- Two most similar clusters fused, yielding $n-1$ clusters.
- Next fusion yields $n-2$ clusters.
- Proceed until single cluster remains.

Algorithm 7: Hierarchical clustering.

1. Begin with n observations and a measure of all $n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.
2. For $i = n, n-1, \dots, 2$:
    a. Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.
    b. Compute the new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.

Unsupervised Learning

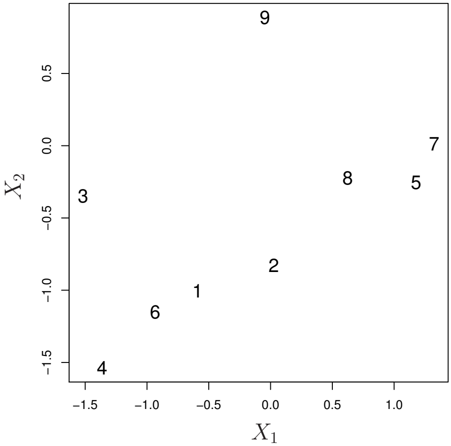
Hierarchical clustering algorithm: linkage

- How is distance measure between groups of observations defined?
- Different notions of **linkage** possible

Linkage	*Description*
Complete	Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *largest* of these dissimilarities.
Single	Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.
Average	Mean intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *average* of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable *inversions*.

Unsupervised Learning

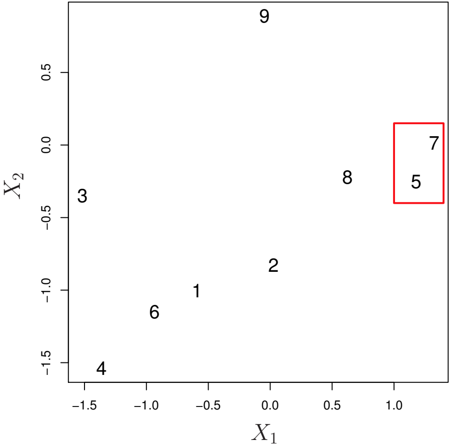
Hierarchical clustering algorithm: linkage



First few steps of hierarchical clustering algorithm of previous data using complete linkage and Euclidean distance.

Unsupervised Learning

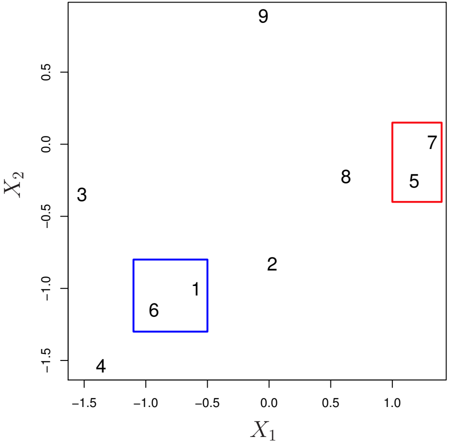
Hierarchical clustering algorithm: linkage



First few steps of hierarchical clustering algorithm of previous data using complete linkage and Euclidean distance.

Unsupervised Learning

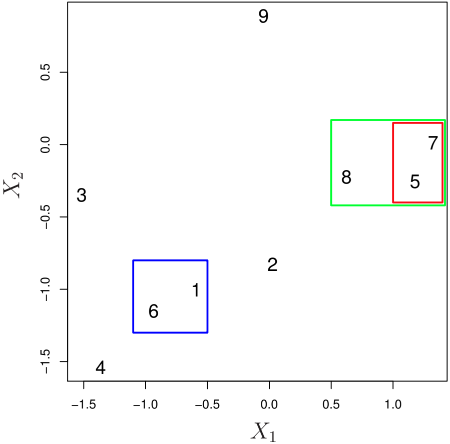
Hierarchical clustering algorithm: linkage



First few steps of hierarchical clustering algorithm of previous data using complete linkage and Euclidean distance.

Unsupervised Learning

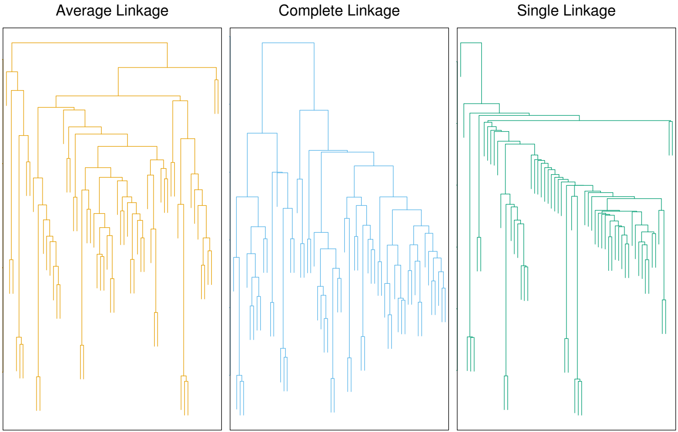
Hierarchical clustering algorithm: linkage



First few steps of hierarchical clustering algorithm of previous data using complete linkage and Euclidean distance.

Unsupervised Learning

Hierarchical clustering algorithm: linkage



Dendrogram resulting from hierarchical clustering algorithm using average, complete and single linkage applied to the same data set. Average and complete linkage tend to produce more balanced dendrograms.

Unsupervised Learning

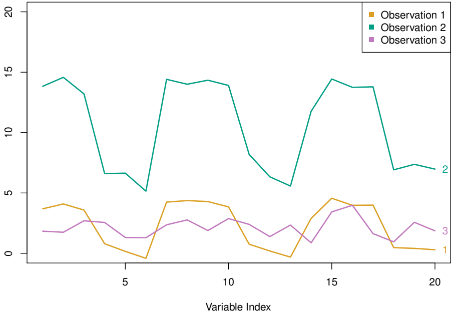
Hierarchical clustering: choice of dissimilarity measure

- Alternative to Euclidean distance: **correlation-based distance**, which considers two observations similar if their features are highly correlated.
- This may be true even if their Euclidean distance is large.

Unsupervised Learning

Hierarchical clustering: choice of dissimilarity measure

- Alternative to Euclidean distance: **correlation-based distance**, which considers two observations similar if their features are highly correlated.
- This may be true even if their Euclidean distance is large.



3 observations of 20 variables each. 1 and 3 have similar values (small Euclidean distance) but are weakly correlated. 1 and 2 have a large Euclidean distance but are closely correlated.

Unsupervised Learning

Hierarchical clustering: choice of dissimilarity measure

Example: Online retailer clustering customers

- Objective: cluster shoppers based on their past shopping histories; identify subgroups of similar shoppers so each group can be shown items/ads of shared interest.
- Data as matrix: rows shoppers, columns items for sale, entries # times shopper has purchased item.
- In Euclidean distance, shoppers who have purchased very few items would be close (may not be desirable).
- In correlation-based distance, shoppers with similar preferences (e.g. who bought items A and B but never C and D) would be close, even if some have purchased in higher volume than others.
- Here correlation-based distance probably better.

Unsupervised Learning

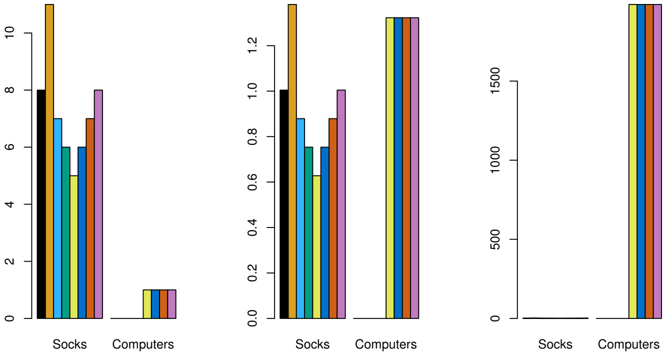
Hierarchical clustering: scaling issues

Scale data to standard deviation one before applying dissimilarity measure?

- Online store again: some items likely purchased more often than others (socks vs. computers).
- High-frequency purchases tend to have stronger effect on inter-shopper dissimilarity.
- Scaling to unit standard deviation before computing inter-observation dissimilarity gives each variable equal importance.
- Also advisable when observation features measured in different scales/units.
- Applies to K-means clustering as well.

Unsupervised Learning

Hierarchical clustering: scaling issues



Online retailer selling only socks and computers. Left: # socks/computers purchased by 8 customers (distinguished by color). In Euclidean-based distance of raw data, computer purchases have little or no effect (less informative, computers have higher margins). Center: each variable scaled by its standard deviation. Right: same data, with *y*-axis showing amount spent on each item.

Unsupervised Learning

Practical issues in clustering

Decisions to make a priori

- Standardize observations/features before measuring similarity? (Centering, scaling)
- For hierarchical clustering:
  - Choice of dissimilarity measure?
  - Choice of linkage?
  - Choice of dendrogram cutting height?
- For K-means: choice of K?

Unsupervised Learning

Practical issues in clustering

Decisions to make a priori

- Standardize observations/features before measuring similarity? (Centering, scaling)
- For hierarchical clustering:
  - Choice of dissimilarity measure?
  - Choice of linkage?
  - Choice of dendrogram cutting height?
- For K-means: choice of K?

Validating obtained clusters

- Have we found meaningful subgroups or only clustered the noise?
- Some proposals for assigning p-values to clusters given in [Hastie et al., 2009]

Further issues

- Sometimes assigning all observations to clusters may be inappropriate.
- Example: most observations belong to small number of (unknown) subgroups. A few observations very different from rest. This presence of outliers which shouldn't be in any cluster can heavily distort the clustering outcome.
- This issue addressed by **mixture models** (soft version of K-means clustering), described in [Hastie et al., 2009].
- Non-robustness to data perturbations: perform clustering on n observations, repeat after randomly removing observations. Often result will strongly differ.
- **Recommendations:** Perform clustering repeatedly with different parameter choices and look for patterns which consistently emerge. Also cluster subsets to obtain sense of robustness. View results not as absolute truth, but as starting point for further investigation.