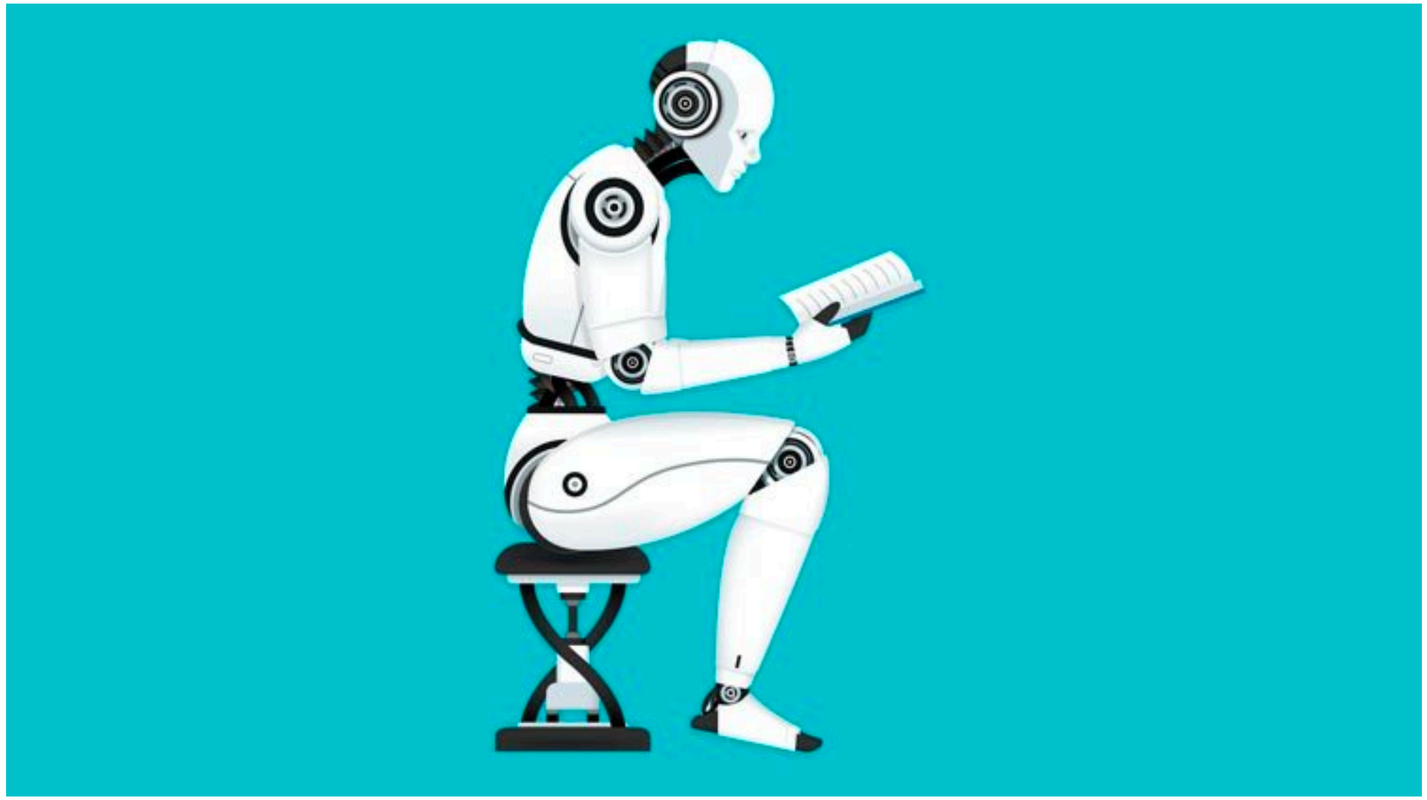
UNIVERSITAT DE BARCELONA



Unsupervised Learning

Machine Learning | Enginyeria Informàtica

Santi Seguí | **2020-2021**

Unsupervised Learning

- Unsupervised vs. Supervised Learning
  - We have mainly focussed on **supervised learning** methods such as regression and classification.
  - In that setting we observe both a set of features $X_1, X_2, \ldots, X_p$ for each object, as well as a response or outcome variable Y . The goal is then to predict Y using $X_1, X_2, \ldots, X_p$.
  - Here we instead focus on **unsupervised learning**, we observe only the features $X_1, X_2, \ldots, X_p$. We are not interested in prediction, because we do not have an associated response variable Y.

Unsupervised Learning

- Unsupervised learning tries to understand the properties of a particular set of data. There are different ways of doing this
  - Clustering - Divide data in groups according to some notion of similarity.
  - Manifold learning - Understanding how data is distributed in the space, parameterising a manifold.

The Goals of Unsupervised Learning

- Several techniques and methods:
  - **principal components analysis**, a tool used for **data visualization** or **data pre-processing** before supervised techniques are applied
  - **autoencoders**, a neural network to learn efficient data codings
  - **clustering**, a broad class of methods for discovering unknown subgroups in data.

The Challenge of Unsupervised Learning

- Unsupervised learning is more subjective than supervised learning, as there is no simple goal for the analysis, such as prediction of a response.
- But techniques for unsupervised learning are of growing importance in a number of fields:
  - subgroups of breast cancer patients grouped by their gene expression measurements,
  - groups of shoppers characterized by their browsing and purchase histories,
  - movies grouped by the ratings assigned by movie viewers.

Another (big) advantage

- It is often easier to obtain **unlabeled data** - from a lab instrument or a computer - than **labeled data**, which can require human intervention.

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, PCA also serves as a tool for data visualization.

Principal Component Analysis: Details

- The first principal component of a set of features $X_1, X_2, \ldots, X_p$ is the normalized linear combination of the features

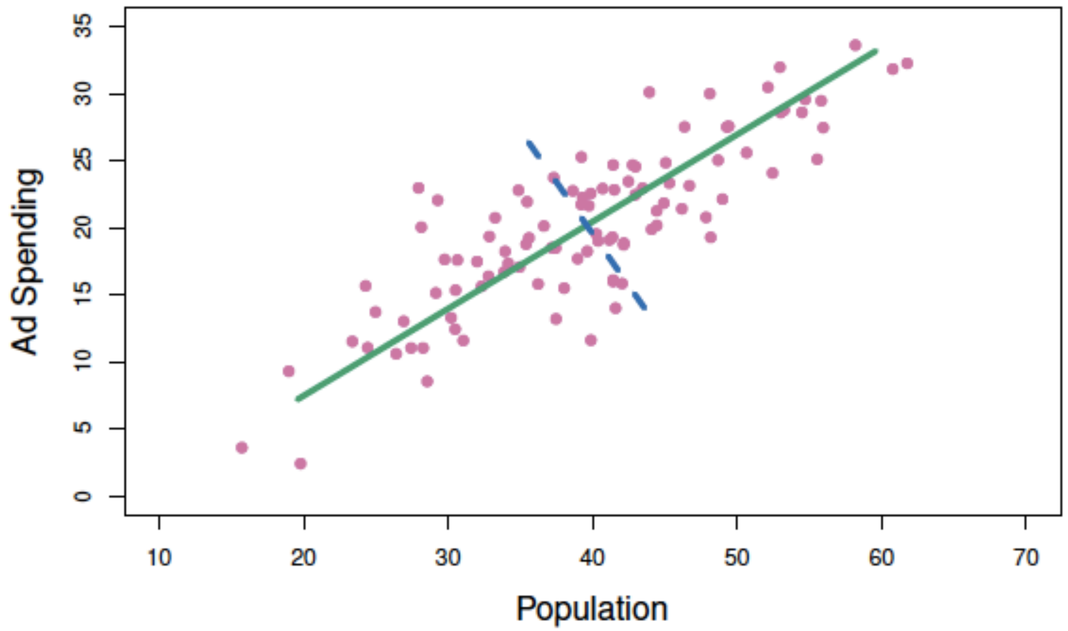
    - $Z_1 = \phi_{11} Z_1 + \phi_{21} Z_2 + \ldots + \phi_{p1} Z_p$

  that has the **largest variance**. By *normalized* we mean that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.

- We refer to the elements $\phi_{11}, \ldots, \phi_{p1}$ as the loadings of the first principal component; together, the loadings make up the principal component loading vector, $\phi_1 = (\phi_{11} \phi_{21} \ldots \phi_{p1})^T$.

- We constrain the loadings so that their sum of squares is equal to one, since otherwise setting these elements to be arbitrarily large in absolute value could result in an arbitrarily large variance.

PCA: example



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.

Computation of Principal Components

- Suppose we have a $n \times p$ data set X . Since we are only interested in variance, we assume that each of the variables in X has been centered to have mean zero (that is, the column means of X are zero).

- We then look for the linear combination of the sample feature values of the form

$$z_{i1} = \phi_{11} x_{i1} + \phi_{21} x_{i2} + \ldots + \phi_{p1} x_{ip}$$

  for $i = 1,...,n$ that has largest sample variance, subject to the constraint that $\sum_{j=1}^{p} \phi_{j1}^2 = 1$.

- Since each of the x_{ij} has mean zero, then so does z_{i1} (for any values of ϕ_{j1}). Hence the sample variance of the z_{i1} can be written as $\frac{1}{n} \sum_{i=1}^{n} z^2$.

Computation: continued

- We can optimization problem as follows

$$maximize_{\phi_{11},\ldots,\phi_{p1}} \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{p}\phi_{j1}x_{ij}\right)^2 subject\ to\ \sum_{j=1}^{p}\phi_{j1}^2 = 1$$

- This problem can be solved via a singular-value decomposition of the matrix X , a standard technique in linear algebra.

- We refer to Z_1 as the first principal component, with realized values$z_{11},\ldots,z_{n1}$.

Geometry of PCA

- The loading vector ϕ_1 with elements $\phi_{11}, \phi_{21}, \ldots, \phi_{p1}$ defines a direction in feature space along which the data vary the most.

- If we project the n data points $x_1, \ldots, x_n$ onto this direction, the projected values are the principal component scores $z_{11}, \ldots, z_{n1}$ themselves.

Further principal components

- The second principal component is the linear combination of $X_1, \ldots, X_p$ that has maximal variance among all linear combinations that are **uncorrelated** with Z_1.

- The second principal component scores $z_{12}, z_{22}, \ldots, z_{n2}$ take the form

$$Z_{i2} = \phi_{12} x_{i1} + \phi_{22} x_{i2} + \ldots + \phi_{p2} x_{ip}$$

where ϕ_2 is the second principal component loading vector, with elements $\phi_{12}, \phi_{22}, \ldots \phi p2$.

Further principal components: continued

- It turns out that constraining Z_2 to be uncorrelated with Z_1 is equivalent to constraining the direction ϕ_2 to be orthogonal (perpendicular) to the direction ϕ_1 . And so on.

- The principal component directions $\phi_1, \phi_2, \phi_3, \ldots$. are the ordered sequence of right singular vectors of the matrix X , and the variances of the components are $\frac{n}{1}$ times the squares of the singular values. There are at most $min(n-1, p)$ principal components.

Illustration

- USAarrests data: For each of the fifty states in the United States, the data set contains the number of arrests per 100;000 residents for each of three crimes: Assault , Murder , and Rape. There is aslo a variable called UrbanPop that is the percent of the population in each state living in urban areas.

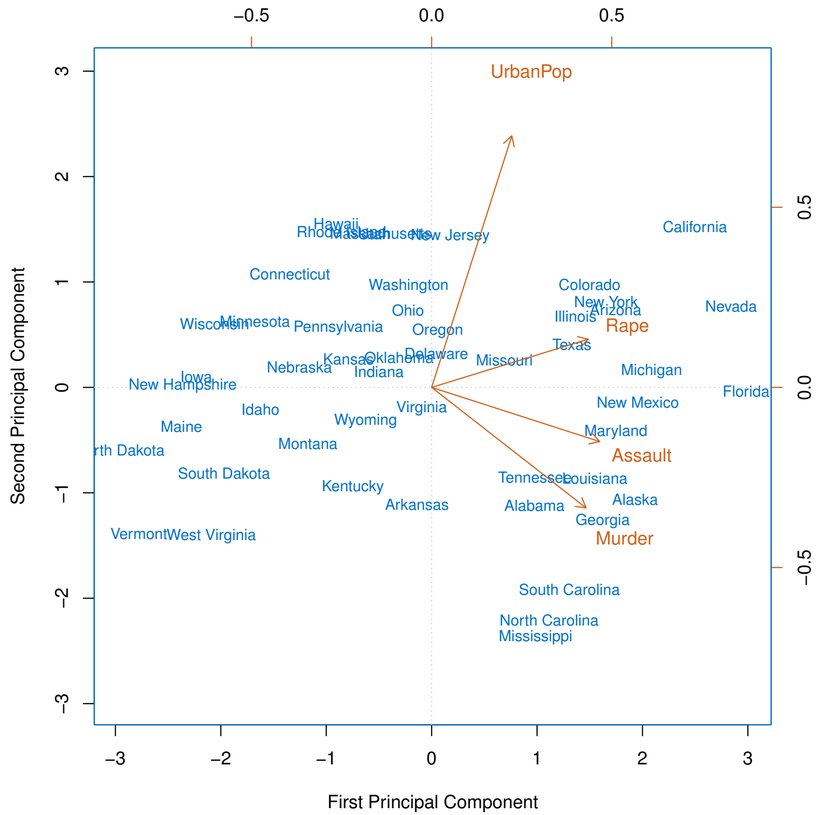
- The principal component score vectors have length $n = 50$, and the principal component loading vectors have length $p = 4$.

- PCA was performed after **standardizing** each variable to have mean zero and standard deviation one.

PCA loadings

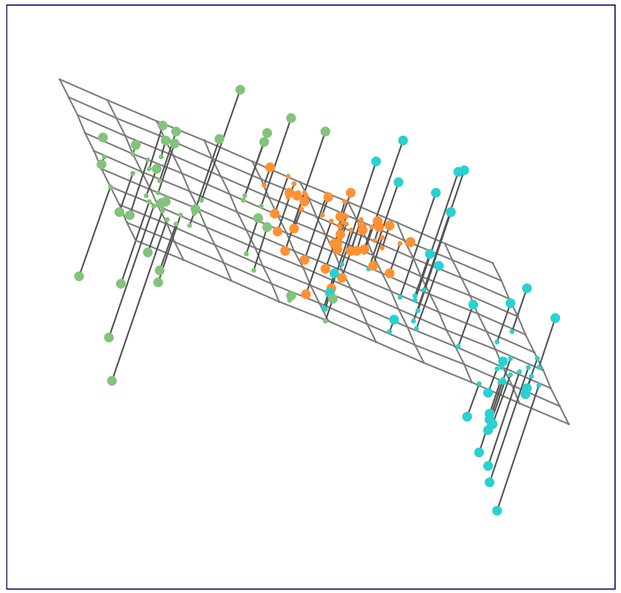
	PC1	PC2
Murder	0.5358995	-0.4181809
Assault	0.5831836	-0.1879856
Urban Pop	0.2781909	0.8728062
Rape	0.5434321	0.1673186

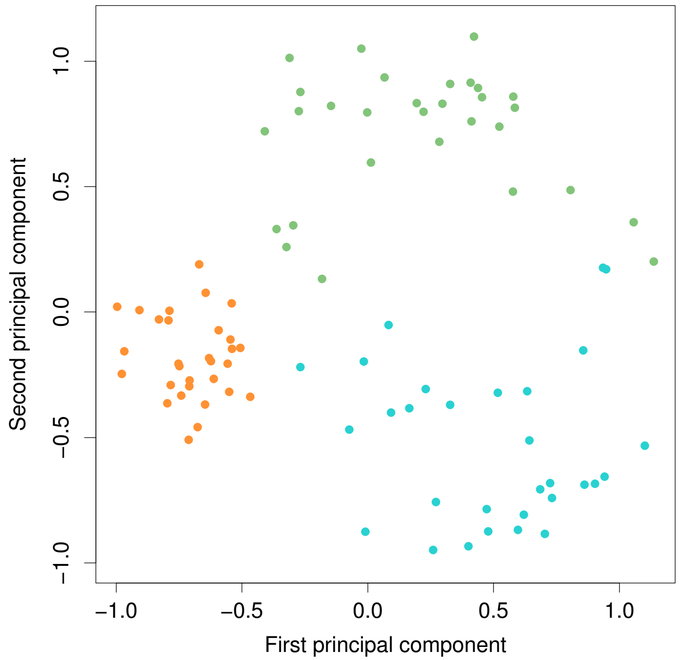
USAarrests data: PCA plot



- The **blue** state names represent the scores for the first two principal components.
- The **orange** arrows indicate the first two principal component loading vectors (with axes on the top and right). For example, the loading for Rape on the first component is 0.54, and its loading on the second principal component 0.17 [the word Rape is centered at the point (0.54, 0.17)].
- This figure is known as a biplot, because it displays both the principal component scores and the principal component loadings

Another Interpretation of Principal Components





PCA finds the hyperplane closest to the observations

- The first principal component loading vector has a very special property: it defines the line in p-dimensional space that is closest to the n observations (using average squared Euclidean distance as a measure of closeness)

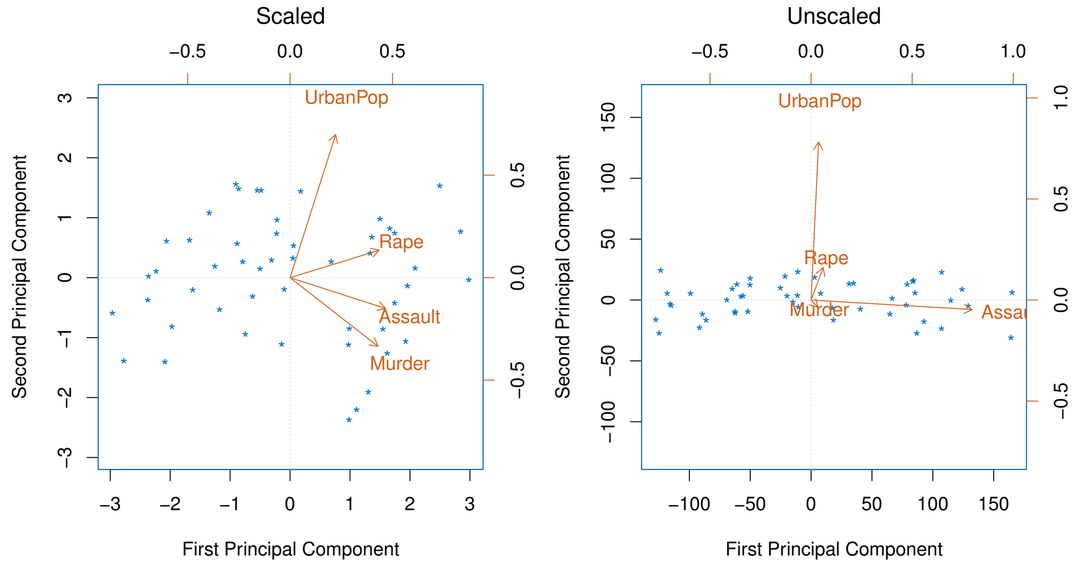
- The notion of principal components as the dimensions that are closest to the n observations extends beyond just the first principal component.

- For instance, the first two principal components of a data set span the plane that is closest to the n observations, in terms of average squared Euclidean distance.

Scaling of the variables matters

- 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.



Proportion Variance explained

- To understand the strength of each component, we are interested in knowing the **proportion of variance explained** (PVE) by each one.

- The total variance present in a data set (assuming that the variables have been centered to have mean zero) is defined as

$$\sum_{j=1}^{p} Var(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2$$

  and the variance explained by the mth principal component is

$$Var(Z_m) = \frac{1}{n} \sum_{i=1}^{n} z_{im}^2$$

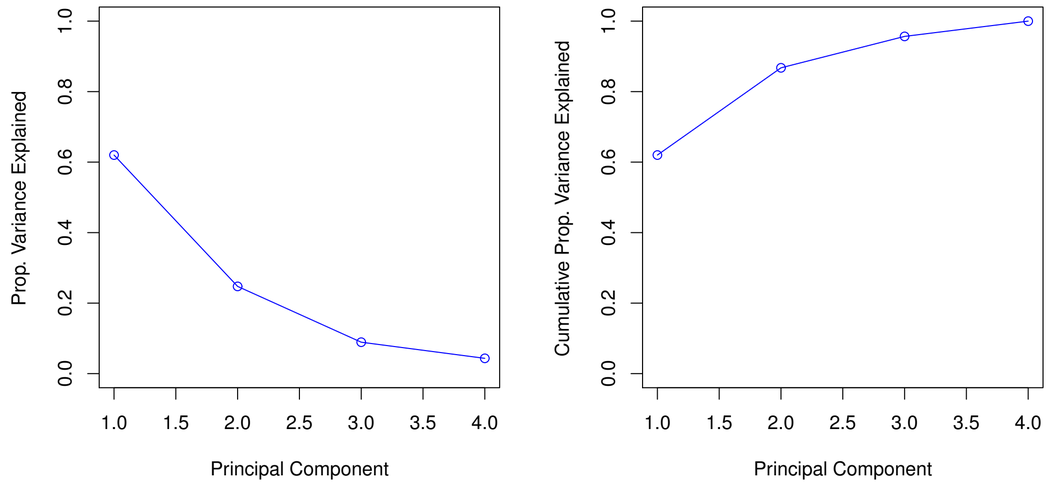
- It can be shown that $\sum_{j=1}^{p} Var(X_j) = \sum_{m=1}^{M} Var(Z_m)$, with $M = min(n-1, p)$.

Proportion Variance Explained: continued

- Therefore, the PVE of the m^{th} principal component is given by the positive quantity between 0 and 1

$$\frac{\sum_{i=1}^{n} z_{im}^2}{\sum_{j=1}^{p} \sum_{i=1}^{n} x_{ij}^2}$$

- The PVEs sum to one. We sometimes display the Variance cumulative PVEs.



How many principal components should we use?

- If we use principal components as a summary of our data, how many components are sufficient?
  - No simple answer to this question, as cross-validation is not available for this purpose.
    - **Why not?**
    - When could we use cross-validation to select the number of components?
  - the "screen plot" on the previous slide can be used as a guide: we look for an "elbow.

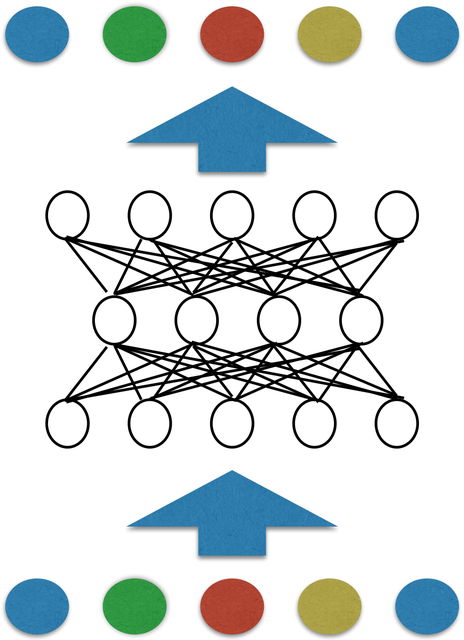
Autoencoders

Autoencoder

- Build a network with the aim of reconstruction.

> D.E. Rumelhart, G.E. Hinton, and R.J. Williams. Learning internal representations by error propagation. In Parallel Distributed Processing. Vol 1: Foundations. MIT Press, Cambridge, MA, 1986.

Autoencoders



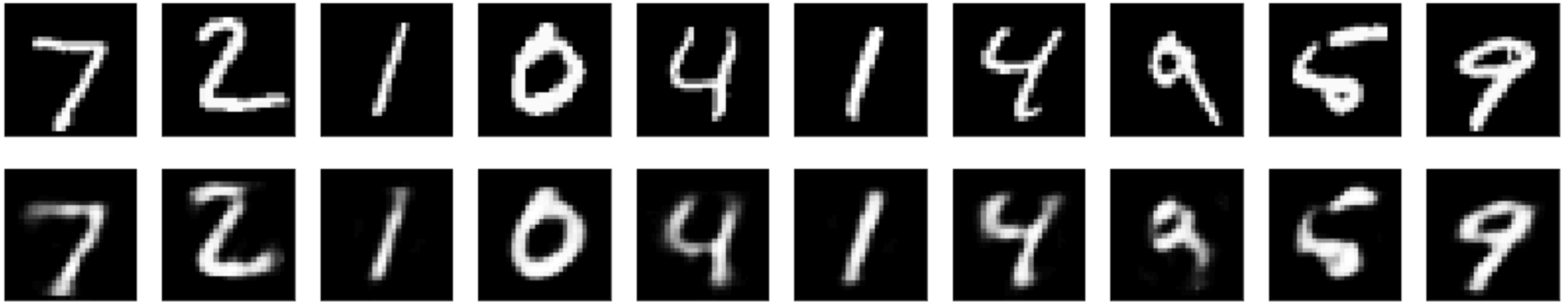
Reconstruction $\tilde{x}$

Loss

$$\mathcal{L}(\tilde{x}, x) = \sum(\tilde{x} - x)^2$$

Input x

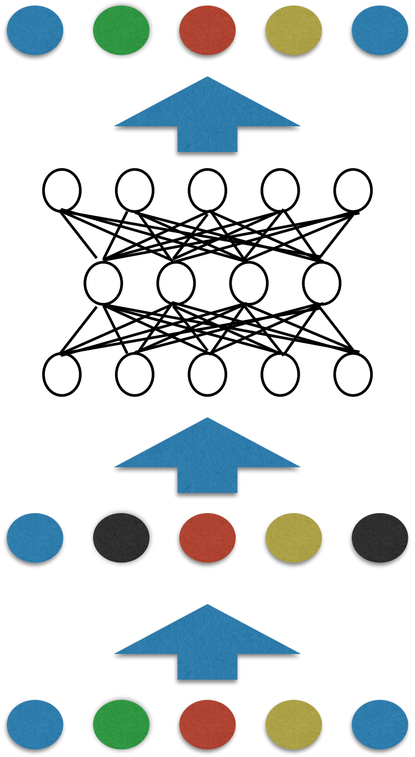
Autoencoders



Problems

- In large networks it may learn the identity mapping rendering the auto encoder representation useless.
- In order to correct this issue and furthermore give robustness to the auto encoder, denonising auto encoders are proposed.

Denoising autoencoders

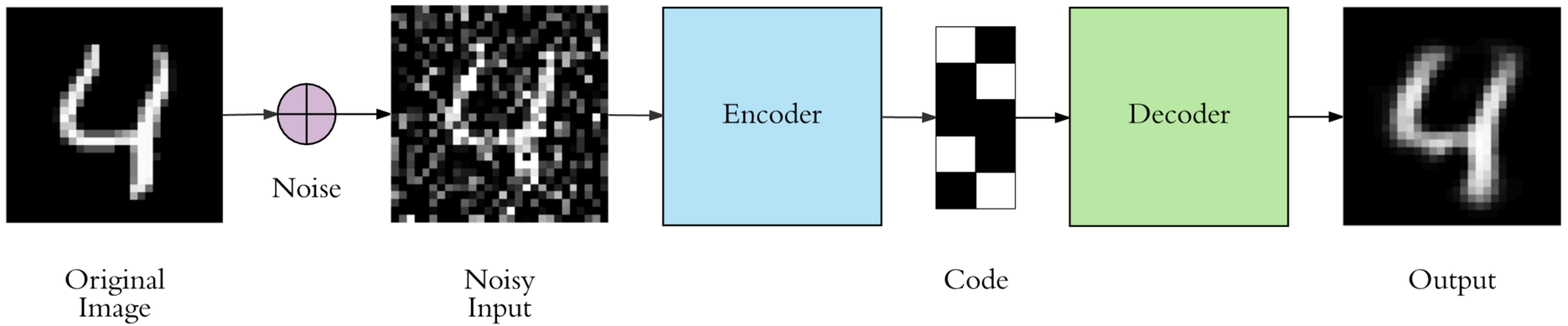


Reconstruction

Noise: dropout input values

Input

Denoising autoencoders



Clustering

Clustering

- Clustering refers to a very broad set of techniques for finding subgroups, or clusters, in a data set.
- We seek a partition of the data into distinct groups so that the observations within each group are quite similar to each other,
- It make this concrete, we must define what it means for two or more observations to be similar or different.
- Indeed, this is often a domain-specific consideration that must be made based on knowledge of the data being studied.

PCA vs. Clustering

- PCA looks for a low-dimensional representation of the observations
- Clustering looks for homogeneous subgroups among the observations.

Clustering for Market segmentation

- Suppose we have access to a large number of measurements (e.g. median household income, occupation, distance from nearest urban area, and so forth) for a large number of people.
- Our goal is to perform market segmentation by identifying subgroups of people who might be more receptive to a particular form of advertising, or more likely to purchase a particular product.
- The task of performing market segmentation amounts to clustering the people in the data set.

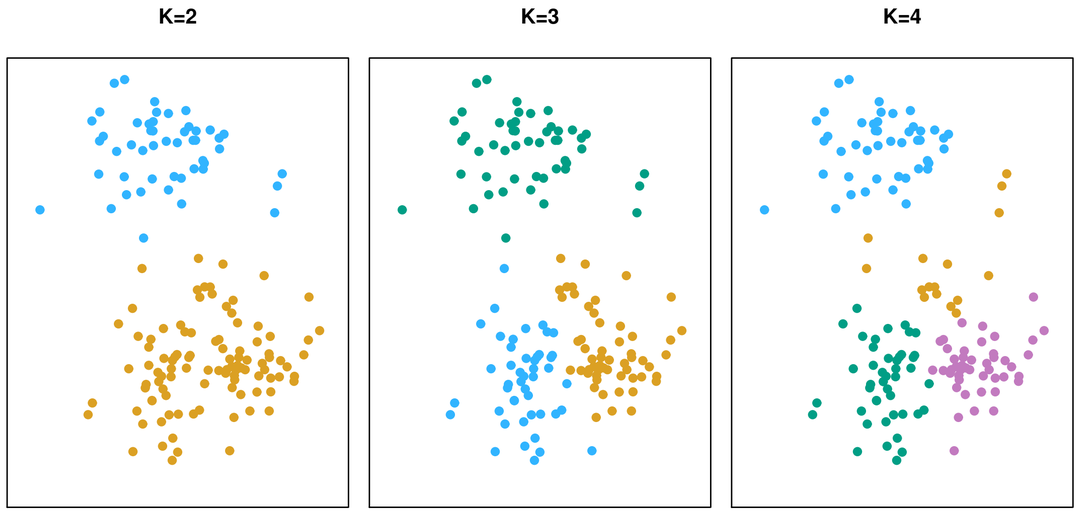
Two clustering methods

- In **K-means clustering**, we seek to partition the observations into a pre-specified number of clusters.

- In **hierarchical clustering**, we do not know in advance how many clusters we want; in fact, we end up with a tree-like visual representation of the observations, called a dendrogram, that allows us to view at once the clusterings obtained for each possible number of clusters, from 1 to n.

K-means

k-means clustering



- A simulated data set with 150 observations in 2-dimensional space. Panels show the results of applying K-means clustering with different values of K, the number of clusters. The color of each observation indicates the cluster to which it was assigned using the K-means clustering algorithm. Note that there is no ordering of the clusters, so the cluster coloring is arbitrary. These cluster labels were not used in clustering; instead, they are the outputs of the clustering procedure.

K-means clustering

- Let $C_1, \ldots, C_K$ denote sets containing the indices of the observations in each cluster. These sets satisfy two properties:
  - $C_1 \cup C_2 \cup \ldots \cup C_K = \{1,...,n\}$. In other words, each observation belongs to at least one of the K clusters.
  - $C_k \cap C_{k'} = \varnothing$ for all $k \neq k'$. In other words, the clusters are non-overlapping: no observations belongs to more than one cluster.
- For instance, if the ith observation is in the kth cluster, the $i \in C_k$.

K-means clustering

- The idea behind K-means clustering is that a **good clustering is one for which the within-cluster variation is as small** as possible.

- The within-cluster variation (WCV) for cluster C_k is a measure $WCV(C_k)$ of the amount by which the observations within a cluster differ from each other.

- Hence we want to solve the problem

$$minimize_{C_1,\ldots,C_K}\left\{\sum_{k=1}^{K} WCV(C_k)\right\}$$

- In words, this formula says that we want to partition the observations into K clusters such that the total within-cluster variation, summed over all K clusters, is as small as possible

How to define within-cluster variation

- Typically we use Euclidean distance

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

- The optimization problem that defines K-means clustering is defined as

$$minimize_{C_1,\ldots,C_K} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

K-means clustering algorithm

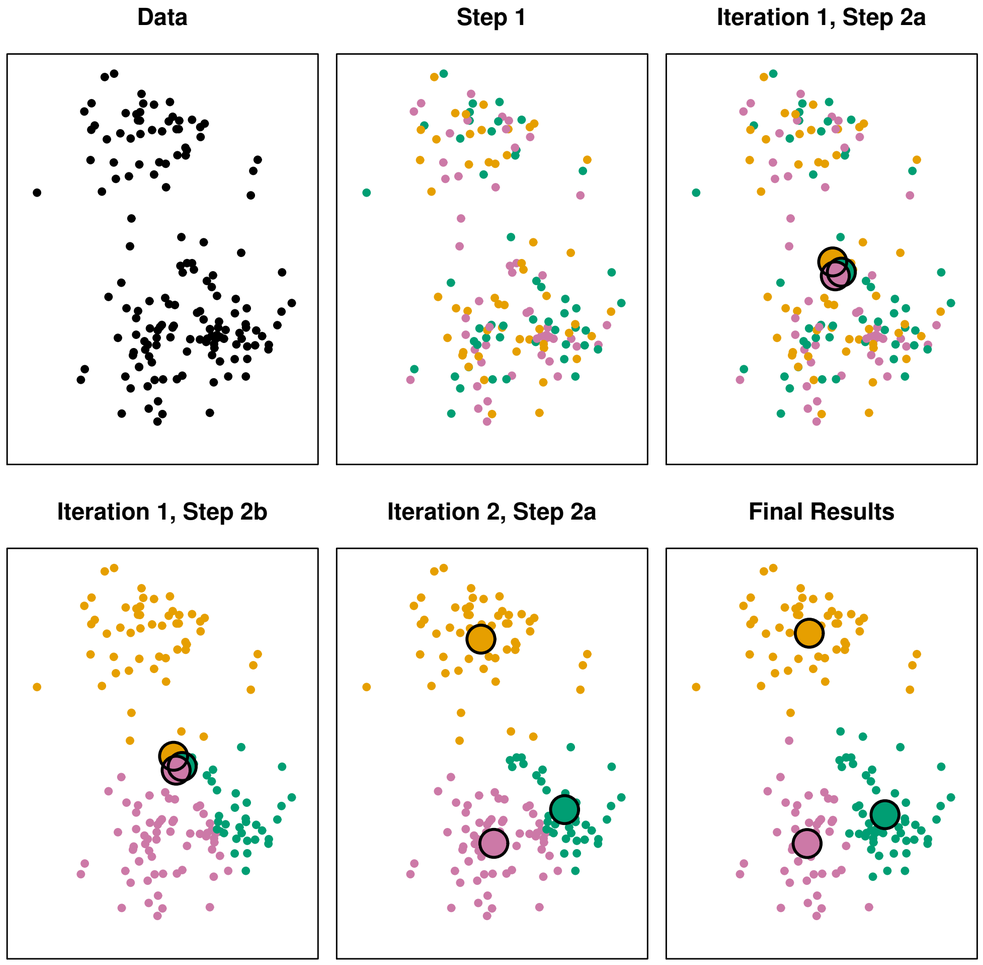
- Randomly assign a number, from 1 to K, to each of the observations. These serve as initial cluster assignments for the observations.

- Iterate until the cluster assignments stop changing:

  - For each of the K clusters, compute the cluster centroid. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.

  - Assign each observation to the cluster whose centroid is closest (where **closest** is defined using Euclidean distance).

Example


Data
Step 1
Iteration 1, Step 2a
Iteration 1, Step 2b
Iteration 2, Step 2a
Final Results

Properties of the Algorithm

- This algorithm is guaranteed to decrease the value of the objective at each step. **Why**?
- Note that

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

- where $\bar{x}_{kj}=\frac{1}{|C_k|}\sum_{i\in C_k}x_{ij}$ is the mean for features j in cluster C_k .
- however it is not guaranteed to give the global minimum. **Why not?**

Properties of the Algorithm

- This algorithm is guaranteed to decrease the value of the objective at each step. **Why**?
- Note that

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

- where $\bar{x}_{kj}=\frac{1}{|C_k|}\sum_{i\in C_k}x_{ij}$ is the mean for features j in cluster C_k .
- however it is not guaranteed to give the global minimum. **Why not?**

Example: different starting values

Hierarchical Clustering

Hierarchical Clustering

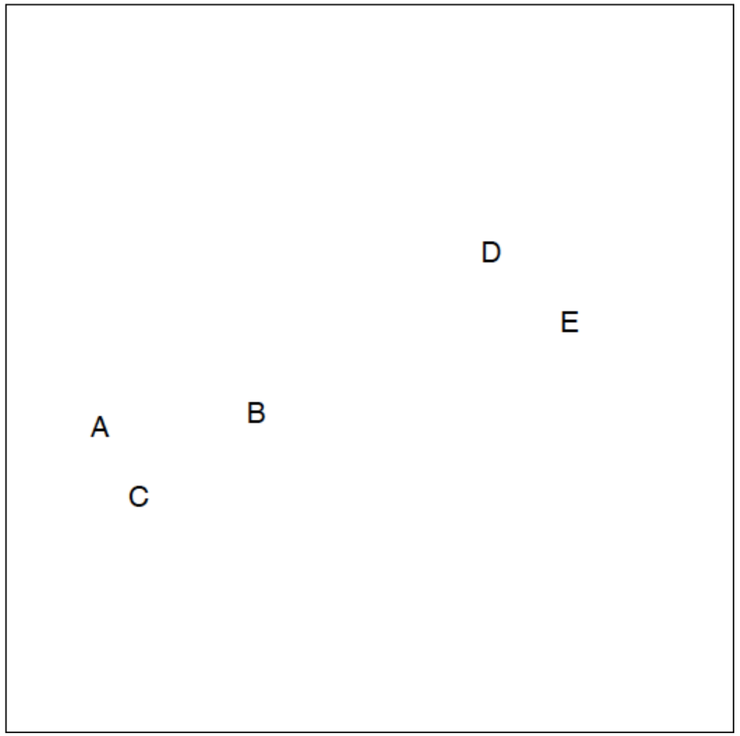
- K-means clustering requires us to pre-specify the number of clusters K. This can be a disadvantage (later we discuss strategies for choosing K)

- Hierarchical clustering is an alternative approach which does not require that we commit to a particular choice of K.

- Now, agglomerative clustering will be described. This is the most common type of hierarchical clustering, and refers to the fact that a dendrogram is built starting from the leaves and combining clusters up to the trunk.

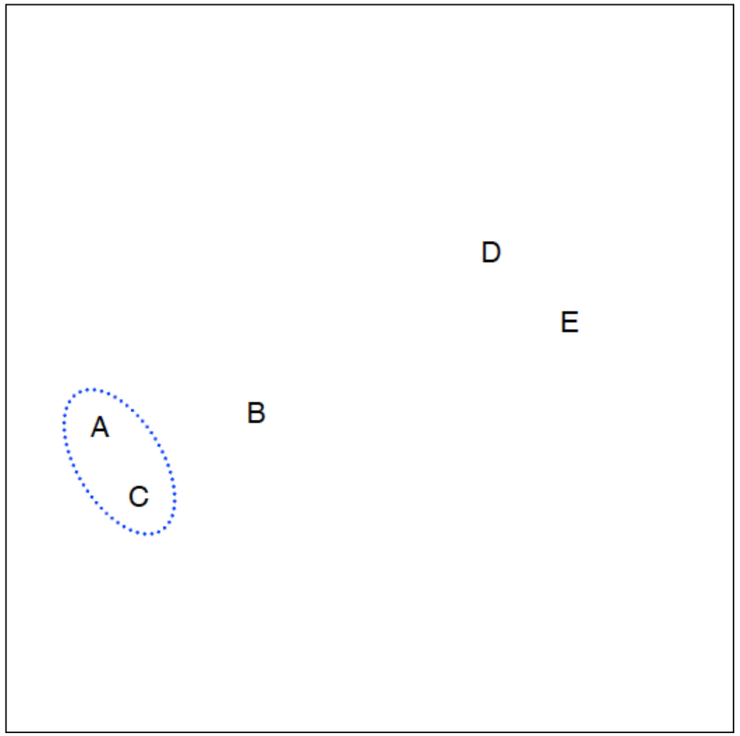
Hierarchical Clustering: the idea

- Builds a hierarchy in a “bottom-up” fashion…



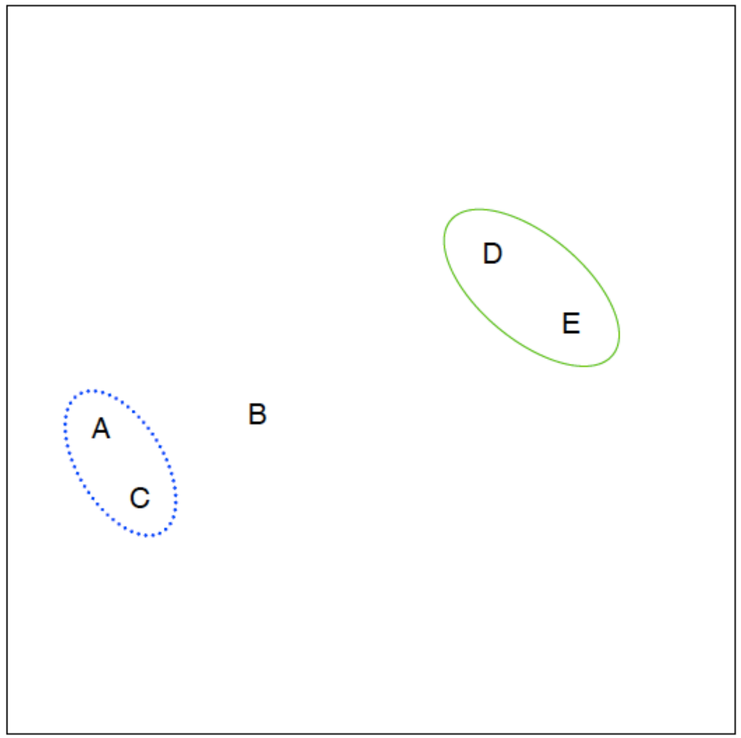
Hierarchical Clustering: the idea

- Builds a hierarchy in a “bottom-up” fashion…



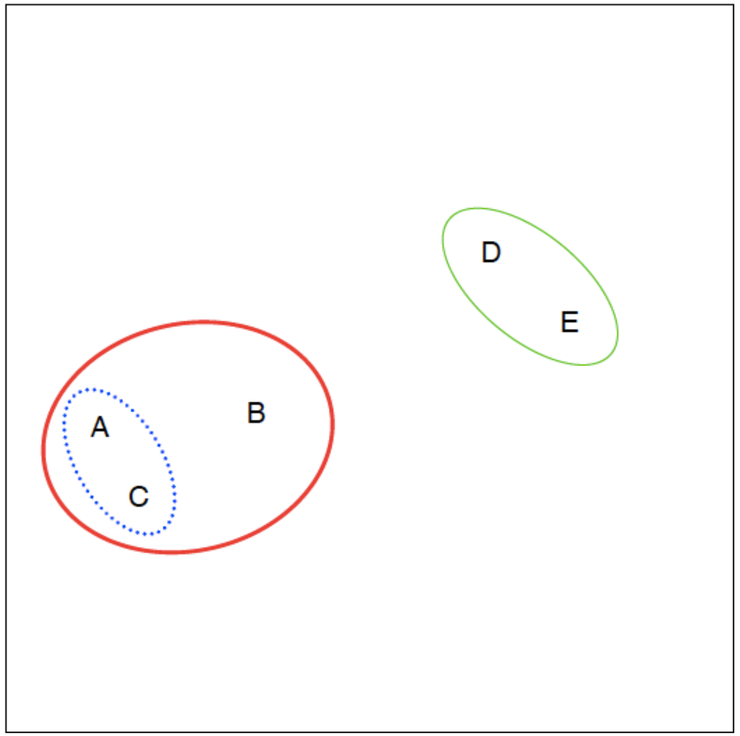
Hierarchical Clustering: the idea

- Builds a hierarchy in a “bottom-up” fashion…



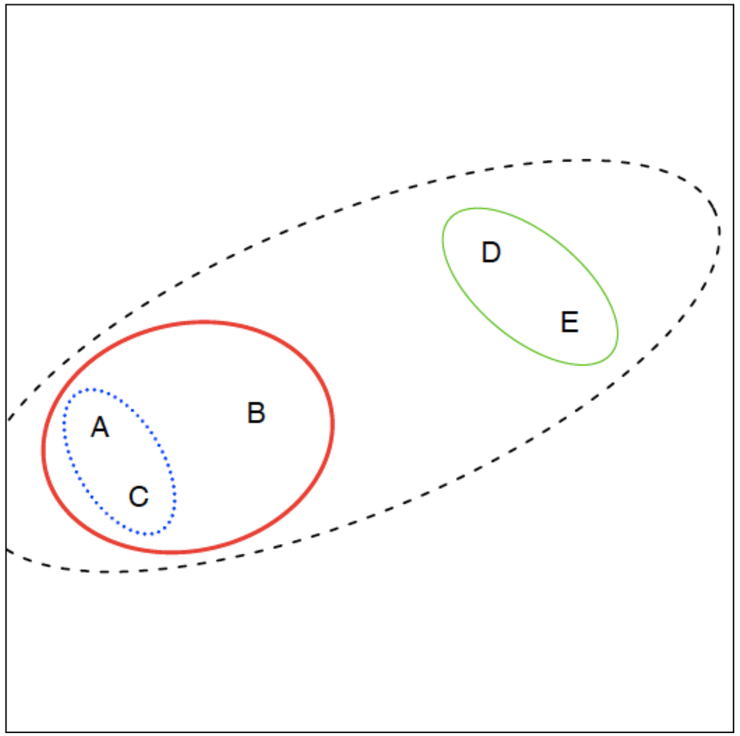
Hierarchical Clustering: the idea

- Builds a hierarchy in a “bottom-up” fashion…



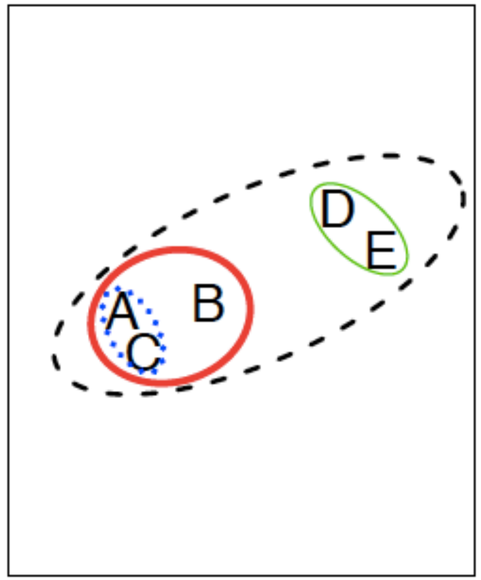
Hierarchical Clustering: the idea

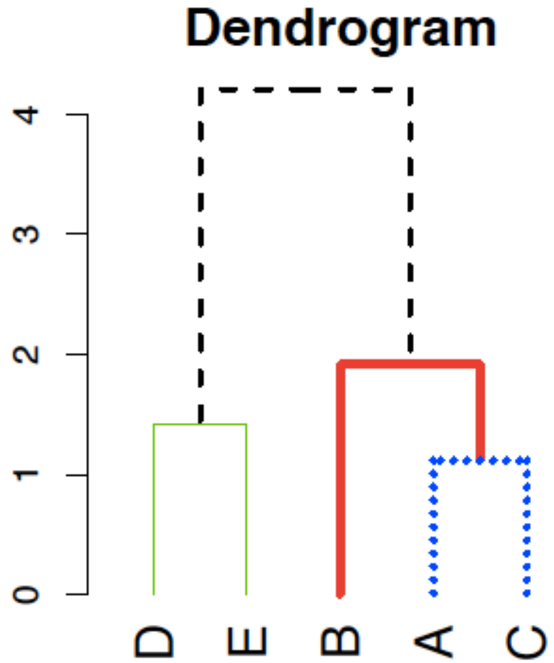
- Builds a hierarchy in a “bottom-up” fashion…



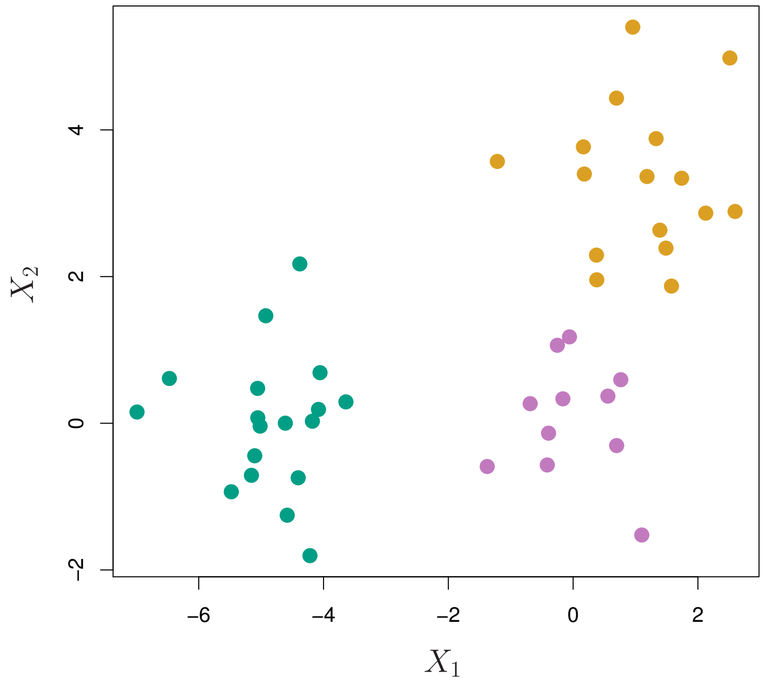
Hierarchical Clustering: the idea

- Builds a hierarchy in a “bottom-up” fashion…



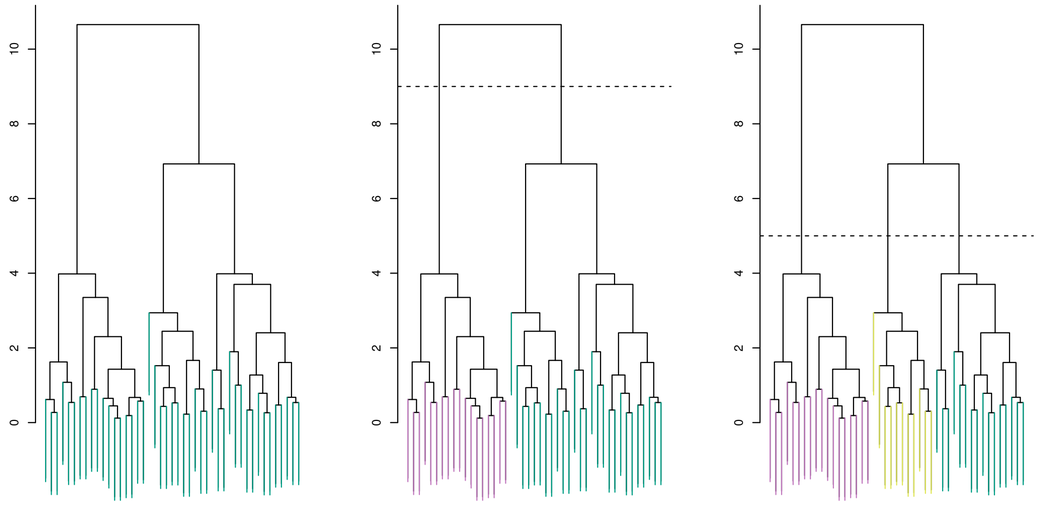


An example



45 observations generated in 2-dimensional space. In reality there are three distinct classes, shown in separate colors. However, we will treat these class labels as unknown and will seek to cluster the observations in order to discover the classes from the data.

Applications of hierarchical clustering

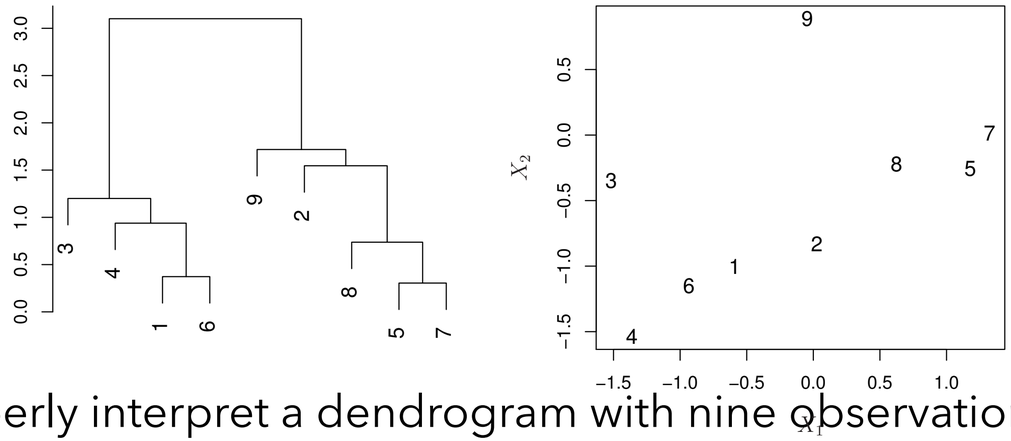


Left: Dendrogram obtained from hierarchically clustering the data from previous slide, with complete linkage and Euclidean distance.

Center: The dendrogram from the left-hand panel, cut at a height of 9 (indicated by the dashed line). This cut results in two distinct clusters, shown in different colors.

Right: The dendrogram from the left-hand panel, now cut at a height of 5. This cut results in three distinct clusters, shown in different colors. Note that the colors were not used in clustering, but are simply used for display purposes in this figure

Another Example



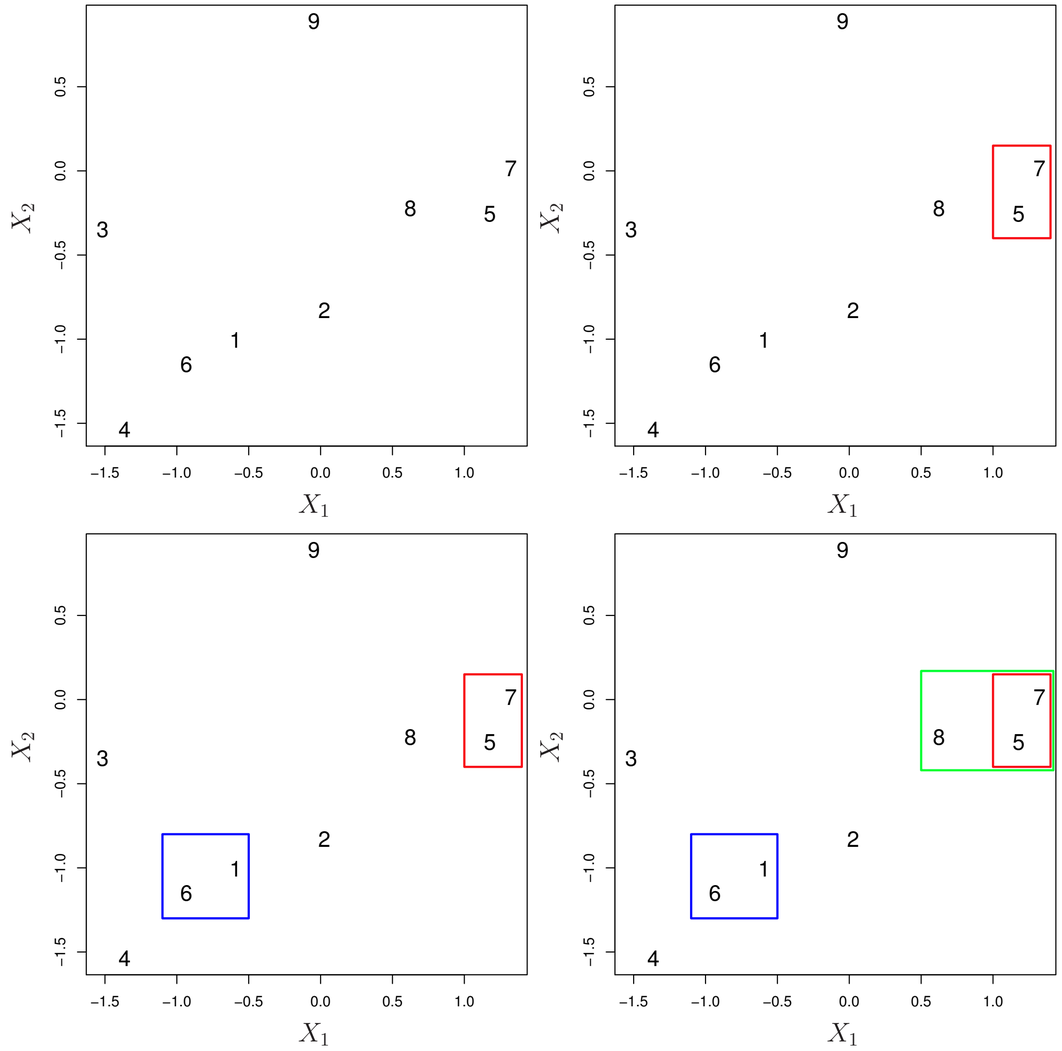
- An illustration of how to properly interpret a dendrogram with nine observations in two-dimensional space. The raw data on the right was used to generate the dendrogram on the left.

- Observations 5 and 7 are quite similar to each other, as are observations 1 and 6.

- However, observation 9 is no more similar to observation 2 than it is to observations 8, 5, and 7, even though observations 9 and 2 are close together in terms of horizontal distance.

- This is because observations 2, 8, 5, and 7 all fuse with observation 9 at the same height, approximately 1.8.

Merges in previous example



Types of linkages

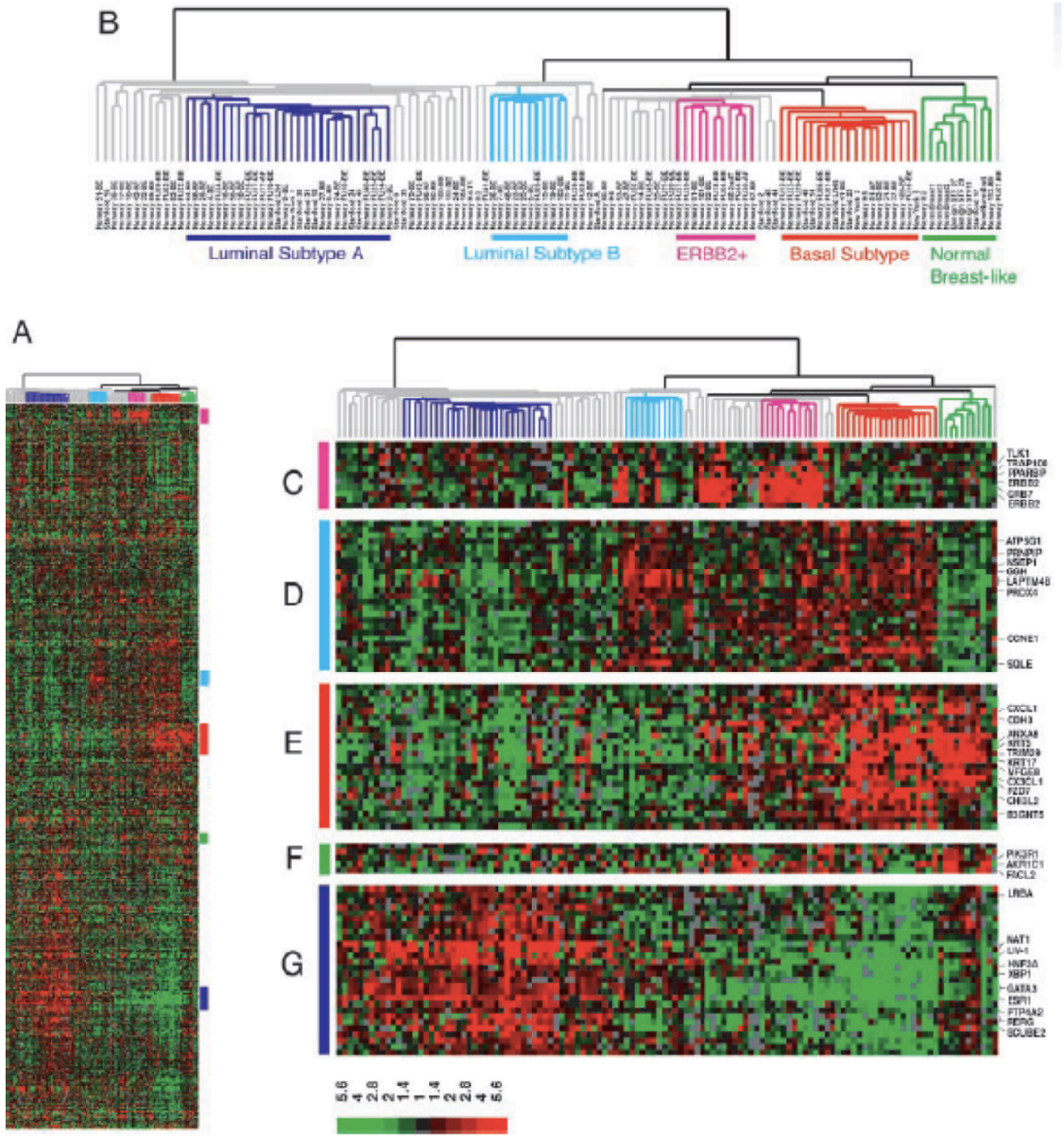
Linkage	Description
Complete	Maximal inter-cluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the largest of these dissimilarities.
Single	Minimal inter-cluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the smallest of these dissimilarities.
Average	Mean inter-cluster 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. Cen- troid linkage can result in undesirable inversions.

Practical Issues

- Should the observations or features first be standardized in some way?For instance, maybe the variables should be centered to have mean zero and scaled to have standard deviation one.
- In the case of hierarchical clustering,
  - What dissimilarity measure should be used?
  - What type of linkage should be used?
- How many clusters to choose? (in both K-means or hierarchical clustering). Difficult problem.

Example: breast cancer microarray study

- “Repeated observation of breast tumor subtypes in independent gene expression data sets;” Sorlie at el, PNAS 2003
- Gene expression measurements for about 8;000 genes, for each of 88 breast cancer patients.
- Average linkage, correlation metric
- Clustered samples using 500 intrinsic genes: each woman was measured before and after chemotherapy. Intrinsic genes have smallest within/between variation


B
Luminal Subtype A
Luminal Subtype B
ERBB2+
Basal Subtype
Normal Breast-like
A
C
D
E
F
G
TLK1
TRAP100
PPARBP
ERBB2
GRB7
ERBB2
ATP5G1
PRNRP
NSEP1
GGH
LAPTM4B
PRDX4
CCNE1
SQLE
CXCL1
CDH3
ANXA8
KRT5
TRIM29
KRT17
MFGE8
CXCL1
FZD7
CNGL2
B3GNT5
PIK3R1
AKR1C1
FACL2
LRRA
NAT1
LIV-1
HNF3A
XBP1
GATA3
ESR1
PTPN12
RERG
SCUBE2
5.6
4.8
2.8
2.1
1.4
1.4
2.8
4.8
5.6

Conclusions

- Unsupervised learning is important for understanding the variation and grouping structure of a set of unlabeled data, and can be a useful pre-processor for supervised learning

- It is intrinsically more difficult than supervised learning because there is no gold standard (like an outcome variable) and no single objective (like test set accuracy)