Module 2. Unsupervised learning: distances and clustering

Philip Moos and Jeff Chang

Outline

- What is unsupervised learning or clustering?
- Distance metrics.
- Clustering Algorithms.
  - k-Means clustering.
  - Hierarchical Clustering.
  - Principal Components Analysis
  - t-SNE

Objectives

- To understand why and when to use clustering.
- To be able to select a proper distance metric based on the characteristics of the data.
- To be able to compare and contrast popular algorithms for clustering.

- Does anyone have an idea what we mean when we say: supervised and unsupervised learning?

What is Unsupervised Learning?

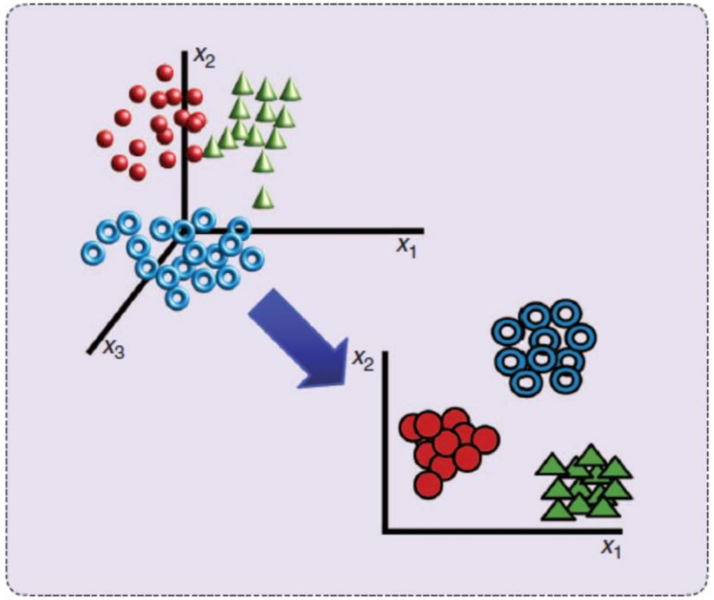
- Unsupervised learning (i.e. clustering) is used to explore the data, and reveal the strongest patterns within the data in an unbiased manner.

- In unsupervised learning, an expectation is that all of the necessary information is within the data and that an appropriate algorithm will cluster the data in an interpretable and meaningful manner leading to the discovery of underlying structure.

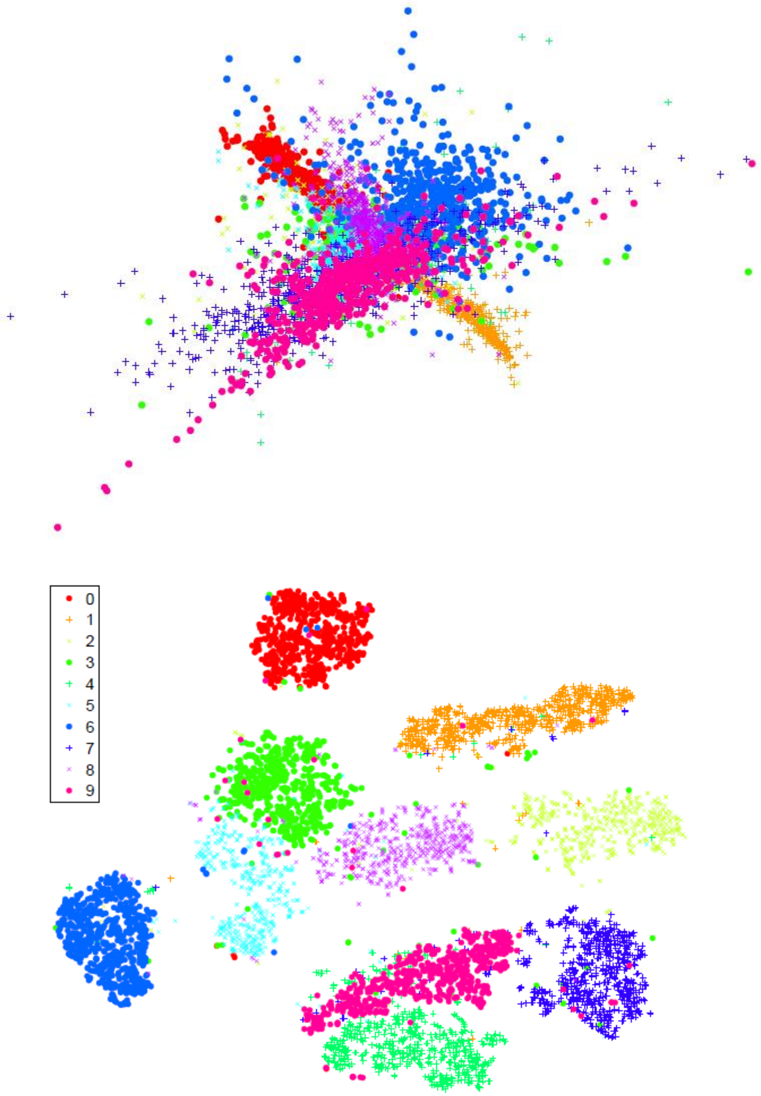
- In unsupervised learning, for each learning task, the expected (or desired) output is not available. Hence, the algorithm has to learn a mapping and estimate the output by itself (without supervision).

Searching for groups

- Clustering is unsupervised or undirected.
- Unlike classification, in clustering, no pre-classified data.
- Search for groups or clusters of data points (records) that are similar to one another.
- Similar points may mean: similar regulation of genes, similar phenotypes, that will behave in similar ways.
- Group points into classes using some distance measures.
  - Within-cluster distance, and between cluster distance
- Applications:
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms



What is good clustering?



A **good clustering** method will produce high quality clusters with

- high **intra-class** similarity
- low **inter-class** similarity

The **quality** of a clustering result depends on

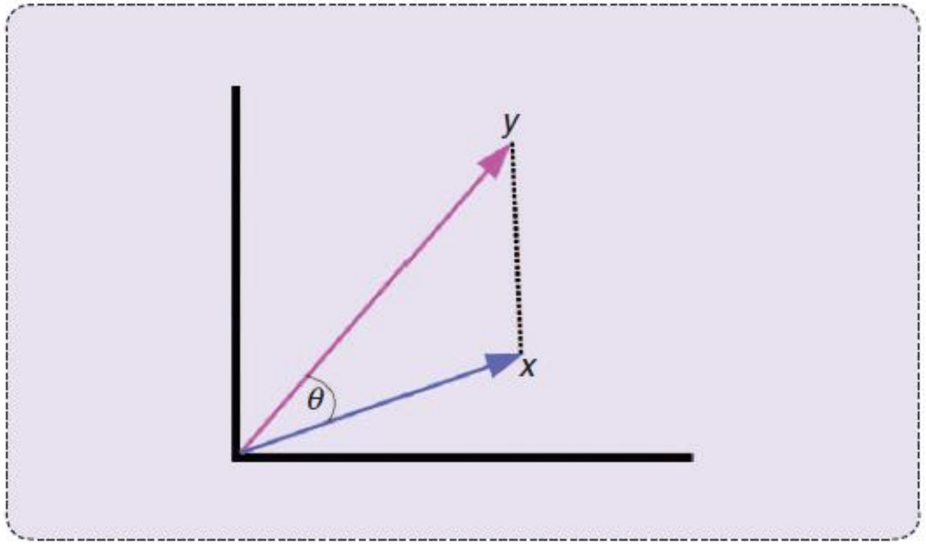
- the similarity measure used
- implementation of the similarity measure

- The **quality** of a clustering method is also measured by its ability to discover some or all of the **hidden** patterns

Desired features of clustering in data mining

- Scalability
- Ability to deal with different types of attributes
- Discovery of clusters with arbitrary shape
- Minimal requirements for domain knowledge to determine input parameters
- Ability to deal with noise and outliers
- Insensitivity to order of input records
- High dimensionality
- Incorporation of user-specified constraints
- Interpretability and usability

Similarity and dissimilarity between objects



- There is **no single definition** of similarity or dissimilarity between data objects
- Similarity/dissimilarity between objects is often expressed in terms of a **distance measure** *d(x,y)*
- Ideally, every distance measure should be a **metric**, i.e., it should satisfy the following conditions:

1. $d(x, y) \geq 0$
2. $d(x, y) = 0 \text{ iff } x = y$
3. $d(x, y) = d(y, x)$
4. $d(x, z) \leq d(x, y) + d(y, z)$

Distance on Numeric Data: Minkowski Distance

- Minkowski distance: A popular distance measure

$$d(i, j) = \sqrt[h]{|x_{i1} - x_{j1}|^h + |x_{i2} - x_{j2}|^h + \cdots + |x_{ip} - x_{jp}|^h}$$

  - where *i* = (x_{i1}, x_{i2}, …, x_{ip}) and *j* = (x_{j1}, x_{j2}, …, x_{jp}) are two p-dimensional data objects, and h is the order (the distance so defined is also called L-*h* norm)

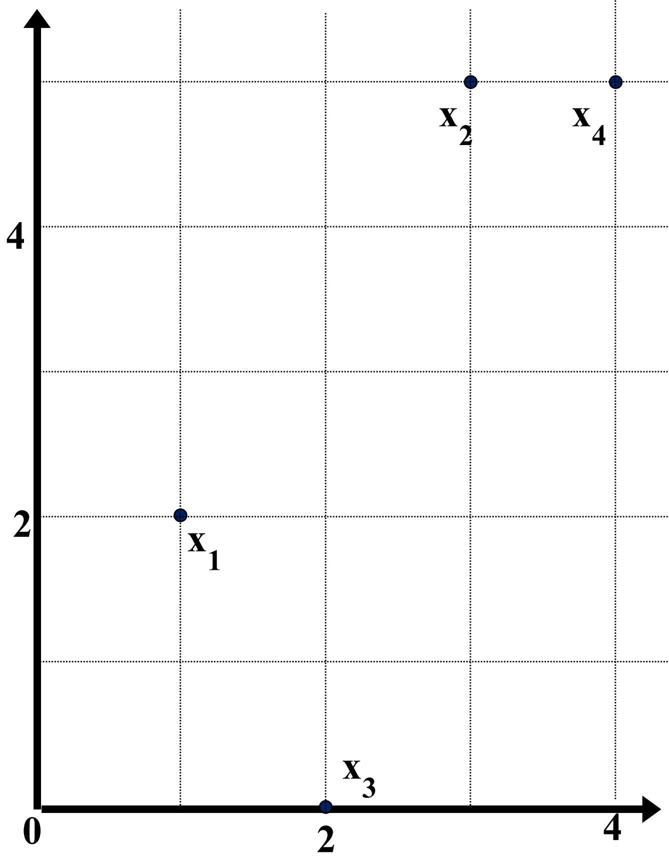
- Note that Euclidean and Manhattan distances are special cases
  - *h* = 1: (L_1 norm) **Manhattan distance**

$$d(i,j) = |x_{i_1} - x_{j_1}| + |x_{i_2} - x_{j_2}| + ... + |x_{i_p} - x_{j_p}|$$

  - *h* = 2: (L_2 norm) **Euclidean distance**

$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + ... + |x_{i_p} - x_{j_p}|^2)}$$

Example: Data Matrix and Distance Matrix



Data Matrix

point	attribute1	attribute2
x1	1	2
x2	3	5
x3	2	0
x4	4	5

Distance Matrix (Manhattan)

	x1	***x2***	***x3***	***x4***
x1	0			
x2	5	0		
x3	3	6	0	
x4	6	1	7	0

Distance Matrix (Euclidean)

	x1	***x2***	***x3***	***x4***
x1	0			
x2	3.61	0		
x3	2.24	5.1	0	
x4	4.24	1	5.39	0

- Any thoughts on when one might want to use Manhattan v. Euclidean distances?

Interval-scaled variables

- **Continuous measurements** of a roughly linear scale
- For example, weight, height and age
- The **measurement unit** can affect the cluster analysis
- To avoid dependence on the measurement unit, we should **standardize** the data

To standardize the measurements:

- calculate the **mean absolute deviation**

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \dots + |x_{nf} - m_f|)$$

where $m_f = \frac{1}{n}(x_{1f} + x_{2f} + \dots + x_{nf})$, and

- calculate the **standardized measurement (*z-score*)**

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

Vector-Based Similarity Measures

- In some situations, distance measures provide a skewed view of data
  - E.g., when the data is very sparse and 0's in the vectors are not significant
  - In such cases, typically vector-based similarity measures are used
  - Most common measure: Cosine similarity

$$\boxed{X = \langle x_1, x_2, \cdots, x_n \rangle} \quad \boxed{Y = \langle y_1, y_2, \cdots, y_n \rangle}$$

  - Dot product of two vectors: $\boxed{sim(X,Y) = X \bullet Y = \sum_i x_i \times y_i}$
  - Cosine Similarity = normalized dot product
  - the norm of a vector X is: $\boxed{\|X\| = \sqrt{\sum_i x_i^2}}$
  - the cosine similarity is: $\boxed{sim(X,Y) = \frac{X \bullet Y}{\|X\| \times \|y\|} = \frac{\sum_i (x_i \times y_i)}{\sqrt{\sum_i x_i^2} \times \sqrt{\sum_i y_i^2}}}$

Vector-Based Similarity Measures

- Why divide by the norm?

$$\boxed{X = \langle x_1, x_2, \cdots, x_n \rangle} \qquad \boxed{\|X\| = \sqrt{\sum_i x_i^2}}$$

- Example:
  - *X* = <2, 0, 3, 2, 1, 4>
  - ||*X*|| = SQRT(4+0+9+4+1+16) = 5.83
  - *X** = *X* / ||*X*|| = <0.343, 0, 0.514, 0.343, 0.171, 0.686>
- Now, note that ||X*|| = 1
- So, dividing a vector by its norm, turns it into a *unit-length* vector
- Cosine similarity measures the angle between two unit length vectors (i.e., the magnitude of the vectors are ignored).

Correlation (Pearson Correlation)

- Correlation measures the linear relationship between objects
- To compute correlation, we standardize data objects, p and q, and then take their dot product

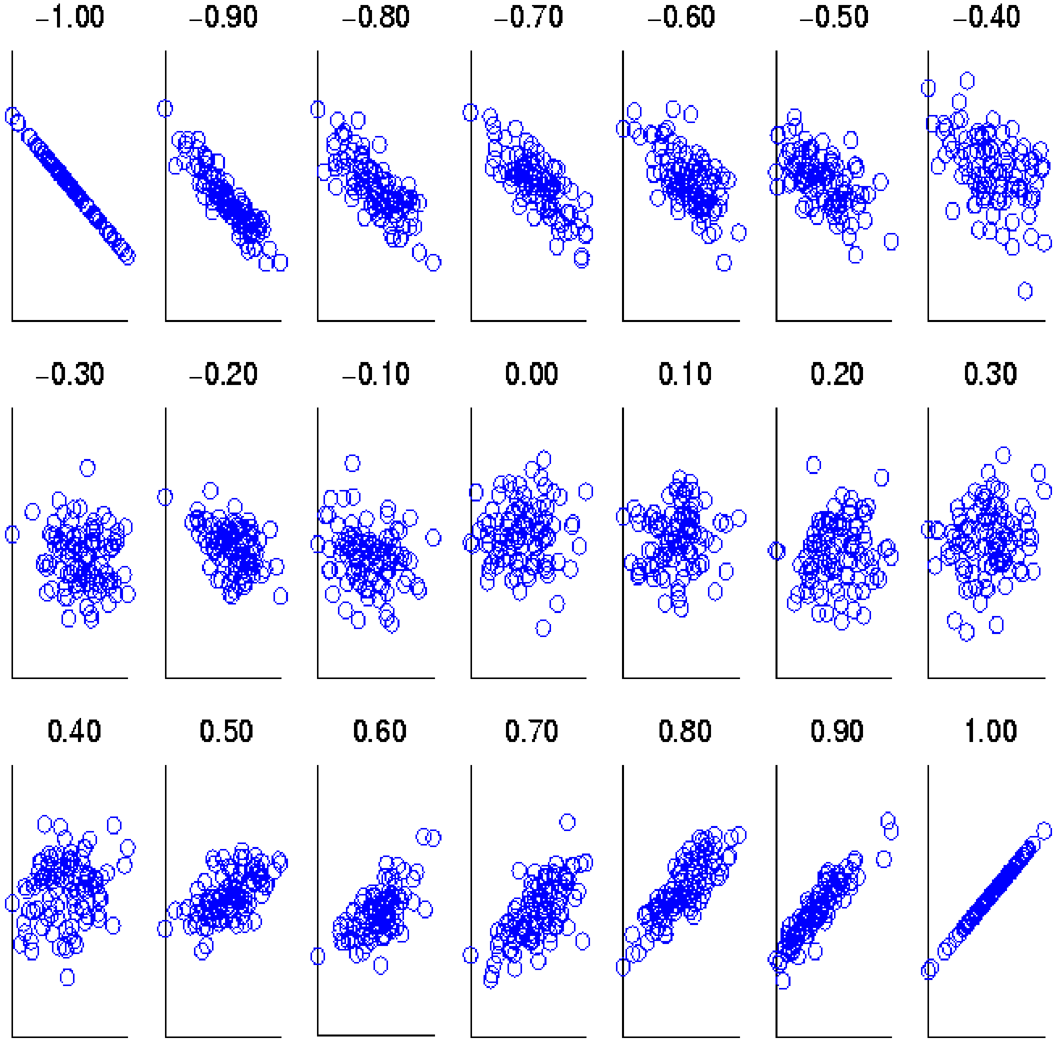
$$p'_k = (p_k - mean(p)) / std(p)$$

$$q'_k = (q_k - mean(q)) / std(q)$$

$$correlation(p, q) = p' \bullet q'$$

Source: "Introduction to Data Mining" by Vipin Kumar et al

Visually Evaluating Correlation



Source: “Introduction to Data Mining” by Vipin Kumar et al

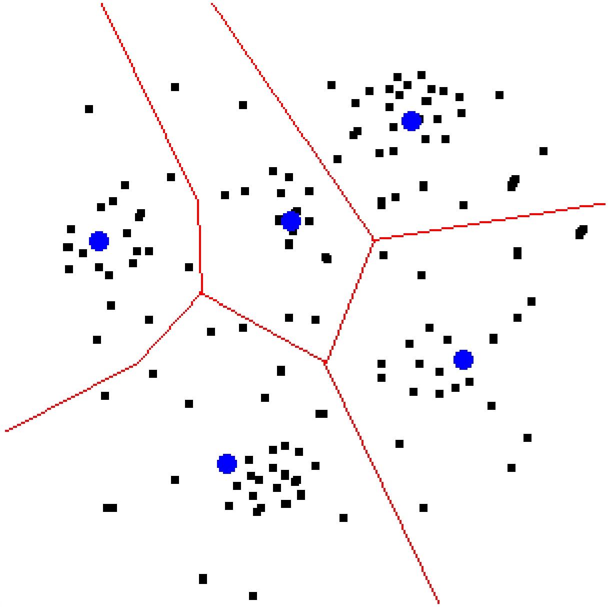
Scatter plots showing the similarity from –1 to 1.

- Does anyone know know the difference between Pearson and Spearman correlation?

Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - k-means : Each cluster is represented by the center of the cluster
  - k-medoids or PAM (Partition around medoids): Each cluster is represented by one of the objects in the cluster

The K-Means Clustering



- Given *k*, the *k-means* algorithm is as follows:
  1) Choose k cluster centers to coincide with k randomly-chosen points
  2) Assign each data point to the closest cluster center
  3) Recompute the cluster centers using the current cluster memberships.
  4) If a convergence criterion is not met, go to 2).

Typical convergence criteria are: no (or minimal) reassignment of data points to new cluster centers, or minimal decrease in squared error.

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} | p - m_i |^2$$

p is a point and m_i is the mean of cluster C_i

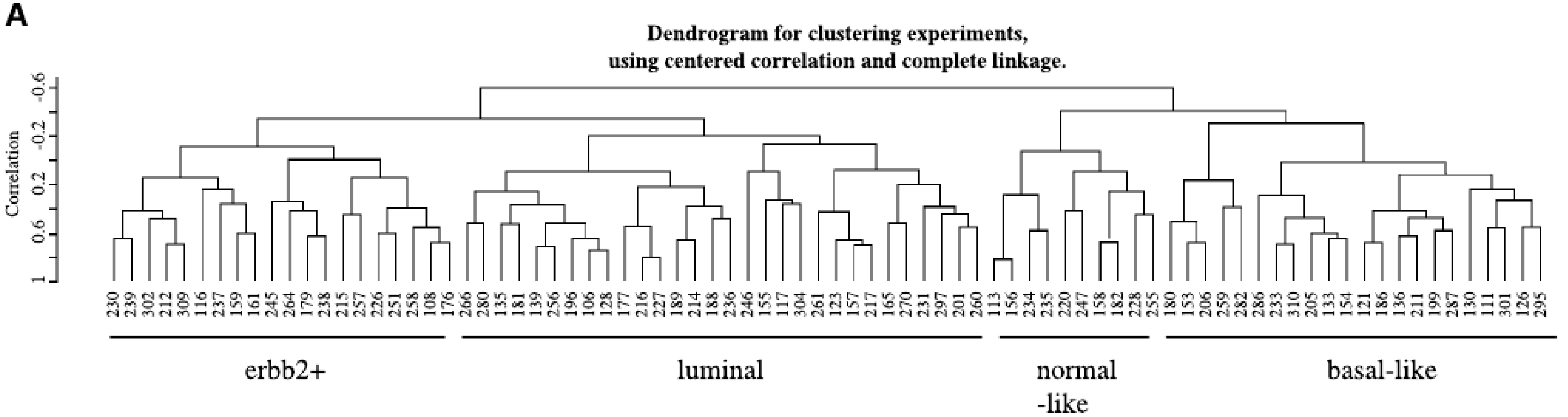
Comments on *K-Means*

- Strength: *efficient*: *O*(*tkn*), where *n* is # data points, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
- Comment: Often terminates at a local optimum. The global optimum may be found using techniques such as: *deterministic annealing* and *genetic algorithms*
- Weakness
  - Applicable only when *mean* is defined, difficult for categorical data
  - Need to specify *k,* the *number* of clusters, in advance
  - Sensitive to noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*
  - Sensitive to initial seeds

Hierarchical methods

- **A hierarchical method:** construct a of clustering is hierarchy of clustering, not just a single partition of objects
- The number of clusters ***k*** is not required as an input
- Use a **distance matrix** as clustering criteria
- A **termination condition** can be used (e.g., a number of clusters)

- The hierarchy often given as a **clustering tree**, also called a **dendrogram**
  - leaves of the tree represent the individual objects
  - internal nodes of the tree represent the clusters

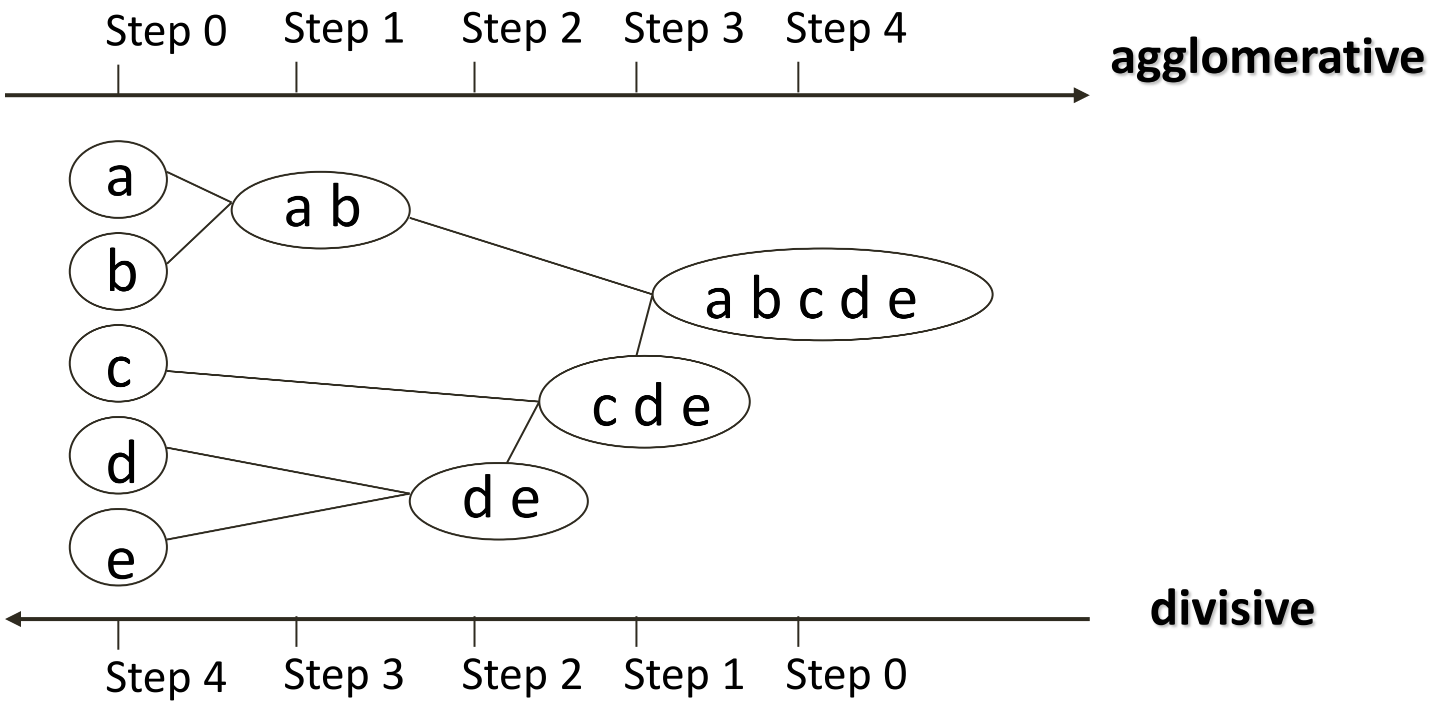


Two types of hierarchical methods

Two main types of hierarchical clustering techniques:

- **agglomerative** (bottom-up):
  - place each object in its own cluster (a singleton)
  - merge in each step the two most similar clusters until there is only one cluster left or the termination condition is satisfied
- **divisive** (top-down):
  - start with one big cluster containing all the objects
  - divide the most distinctive cluster into smaller clusters and proceed until there are *n* clusters or the termination condition is satisfied

Two types of hierarchical methods



Inter-cluster distances

- Three widely used ways of defining the **inter-cluster distance**, i.e., the distance between two separate clusters, are
  - **single linkage method** (nearest neighbour):

    $$d(i, j) = \min_{x \in C_i, y \in C_j} \{ d(x, y) \}$$

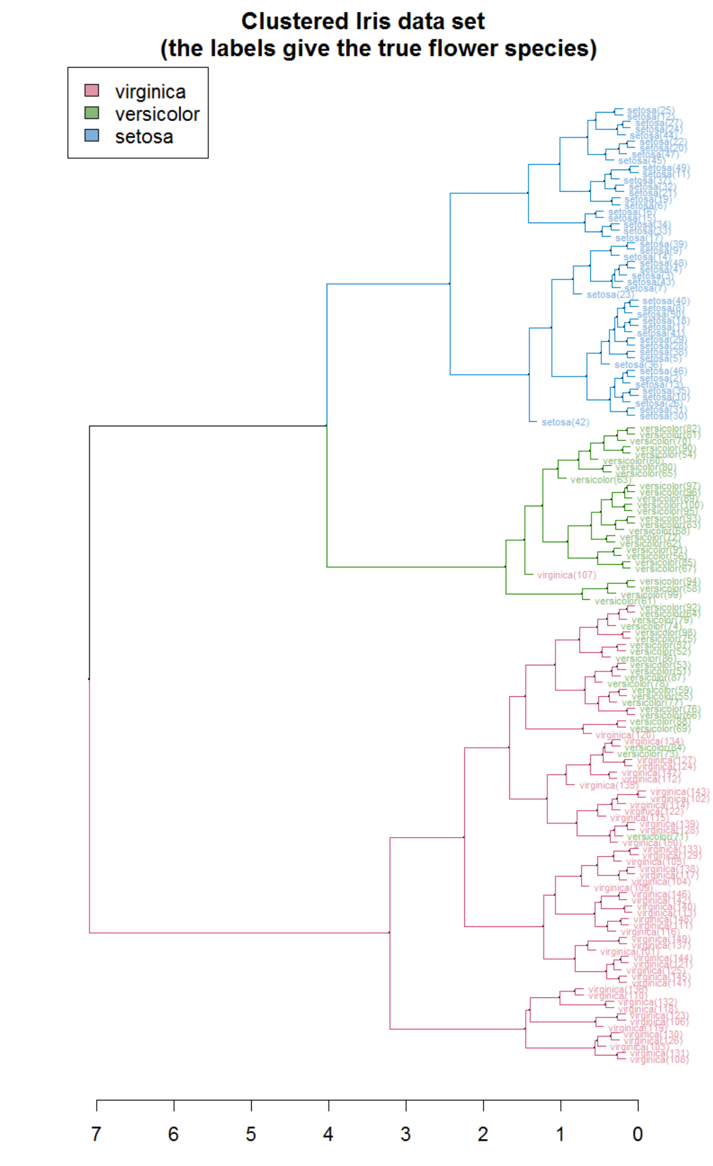
  - **complete linkage method** (furthest neighbour):

    $$d(i, j) = \max_{x \in C_i, y \in C_j} \{ d(x, y) \}$$

  - **average linkage method** (unweighted pair-group average):

    $$d(i, j) = avg_{x \in C_i, y \in C_j} \{ d(x, y) \}$$

Strengths of hierarchical methods

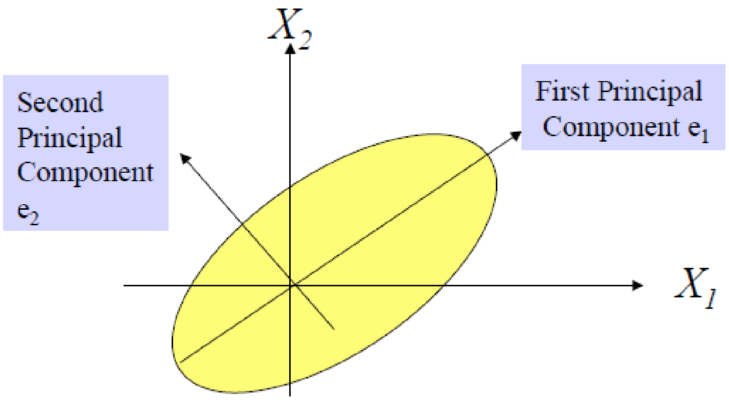


- **Conceptually simple**
- **Theoretical properties** are well **understood**
- When clusters are merged/split, **the decision is permanent => the number of different alternatives** that need to be examined is **reduced**

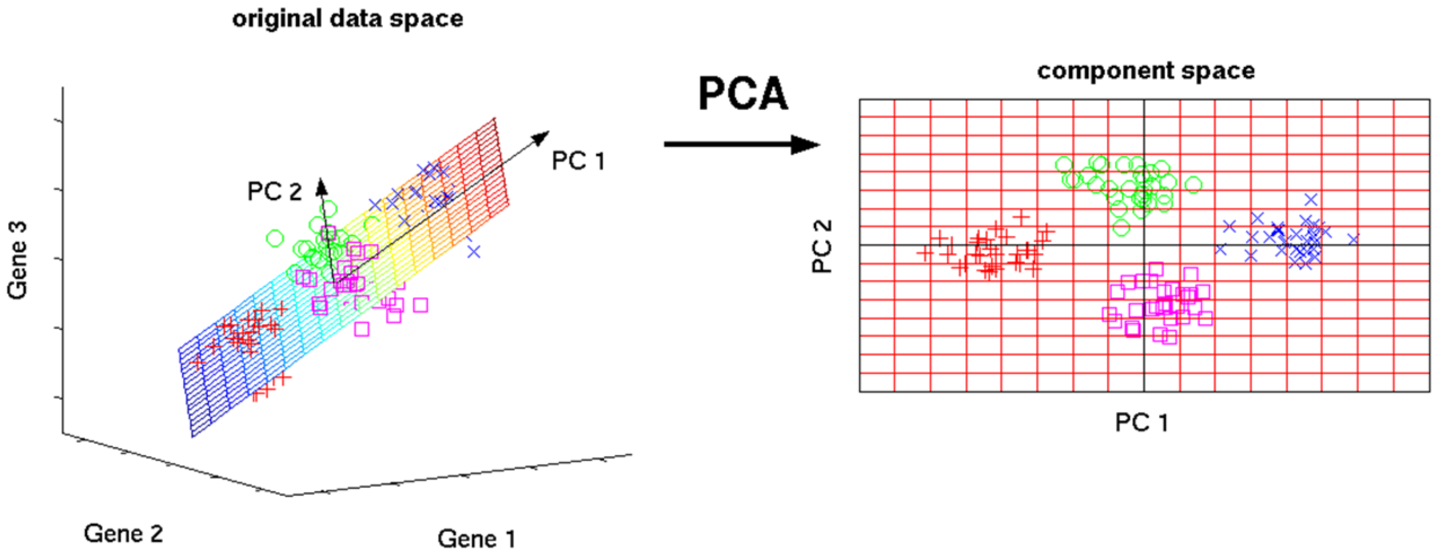
- If you could only choose between K-means and Hierarchical methods, which would you choose, and why?

Principle component analysis

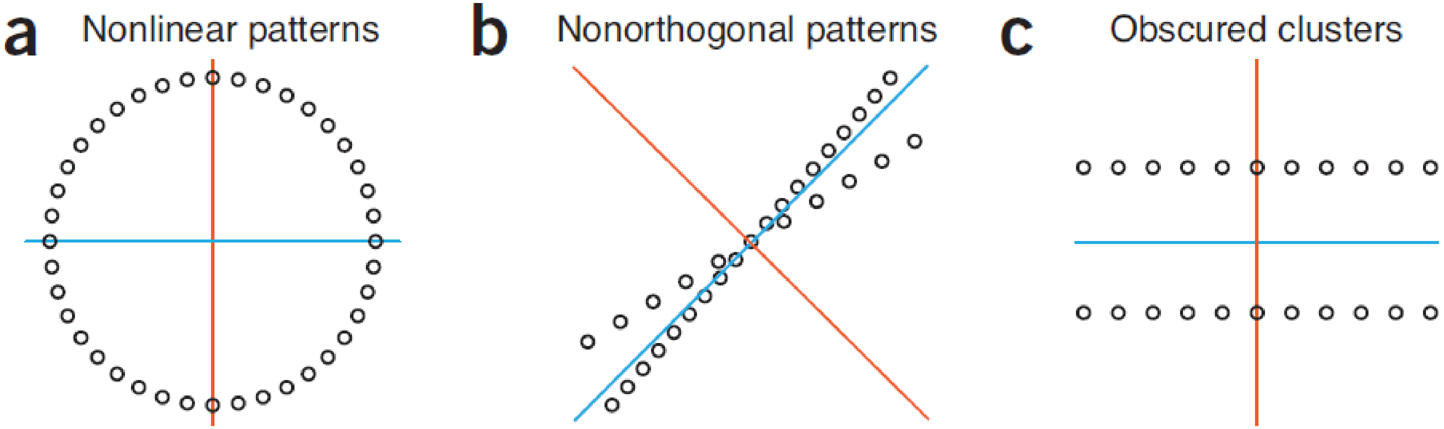
- There will be a whole module on this topic
- Essentially, the goal is to identify the axes where the majority of the variance in the data is located and represent the data on those axes
- Principle components are the eigenvectors of the dataset



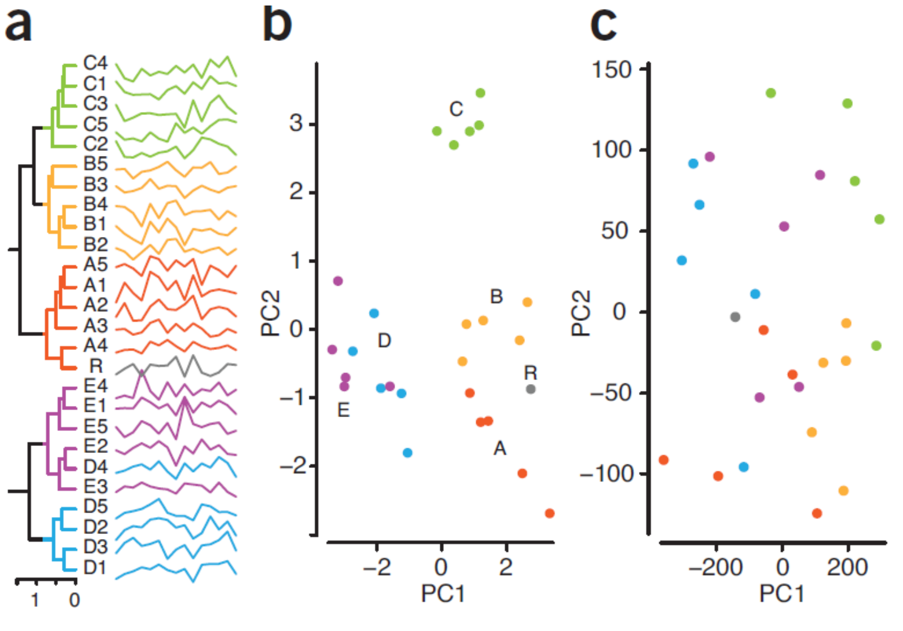
X_2
Second Principal Component e_2
First Principal Component e_1
X_1



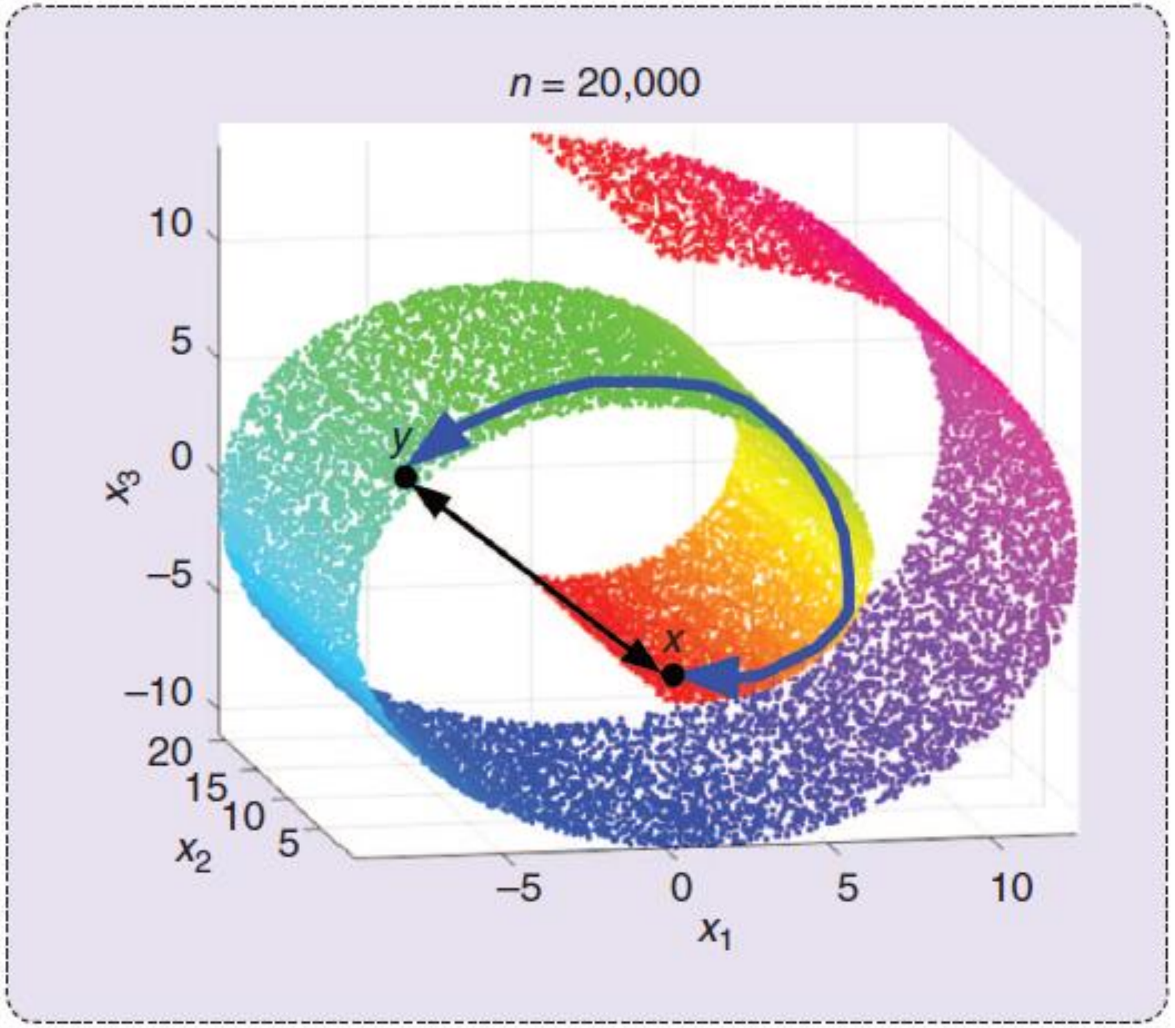
Properties of PCA



- Weaknesses
  - Limited to linear projections
  - Not scale invariant

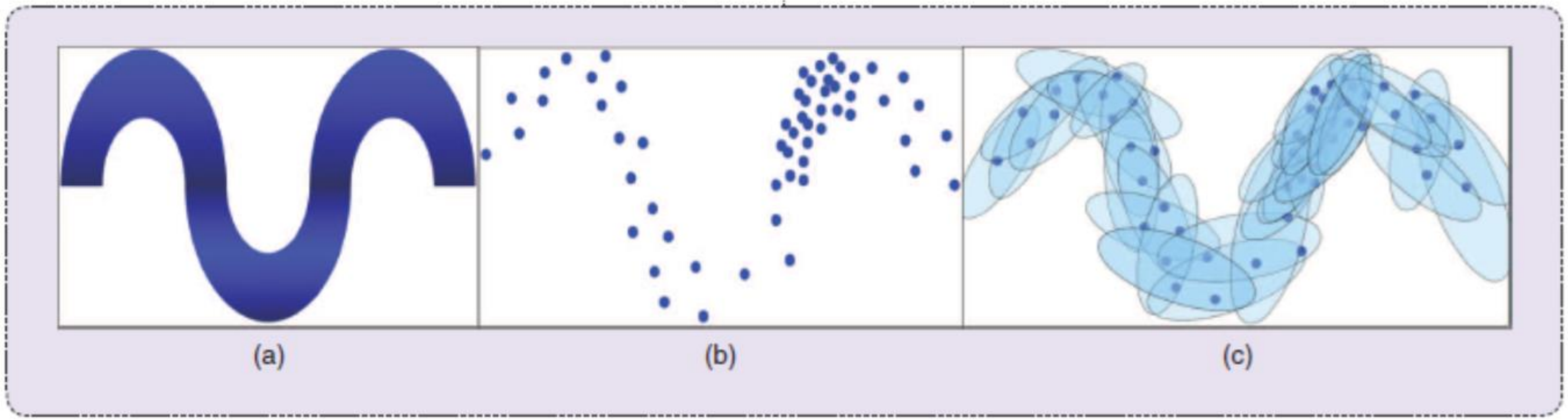


Challenges of non-linear data structures



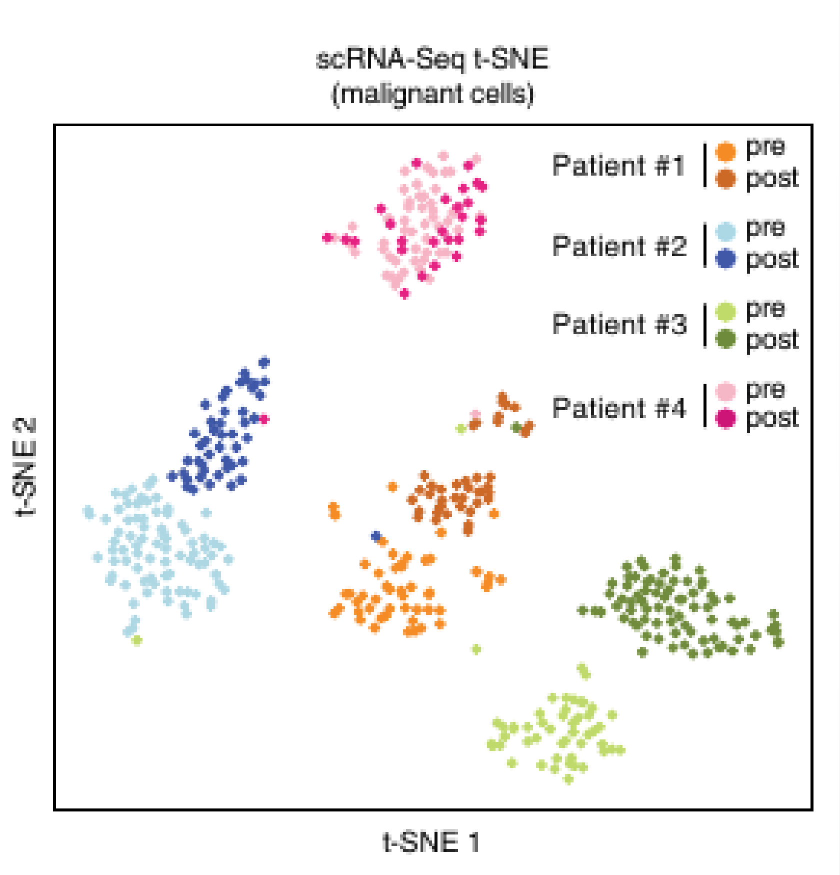
Stochastic neighbor embedding

- Stochastic Neighbor Embedding (SNE) starts by converting the high-dimensional Euclidean distances between datapoints into conditional probabilities that represent similarities using a Gaussian distribution.
- For the low-dimensional counterparts to high-dimensional datapoints, it is possible to compute a similar conditional probability.
- A cost function then finds the samples that are “close”.



t-SNE: t-distributed stochastic neighbor embedding

- A student-t distribution rather than a Gaussian to compute the similarity between two points *in the low-dimensional space*.
- t-SNE employs a heavy-tailed distribution in the low-dimensional space to alleviate both the crowding problem and the optimization problems
- Goal of t-SNE – preserve the higher order structure in the low dimensional space.



Summary

- Unsupervised learning methods can provide a method to evaluate complex datasets
    - As a stand-alone tool to get insight into data distribution
    - As a preprocessing step for other algorithms
- We highlighted a few Distance metrics & Clustering Algorithms.
    - k-Means clustering.
    - Hierarchical Clustering.
    - Principal Components Analysis
    - t-SNE