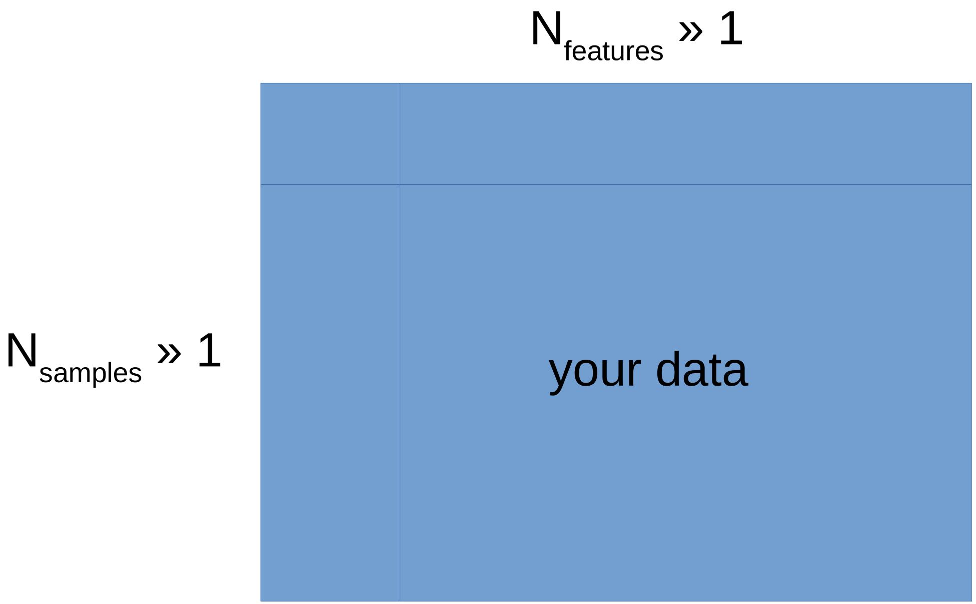
DATA ANALYSIS

Outline

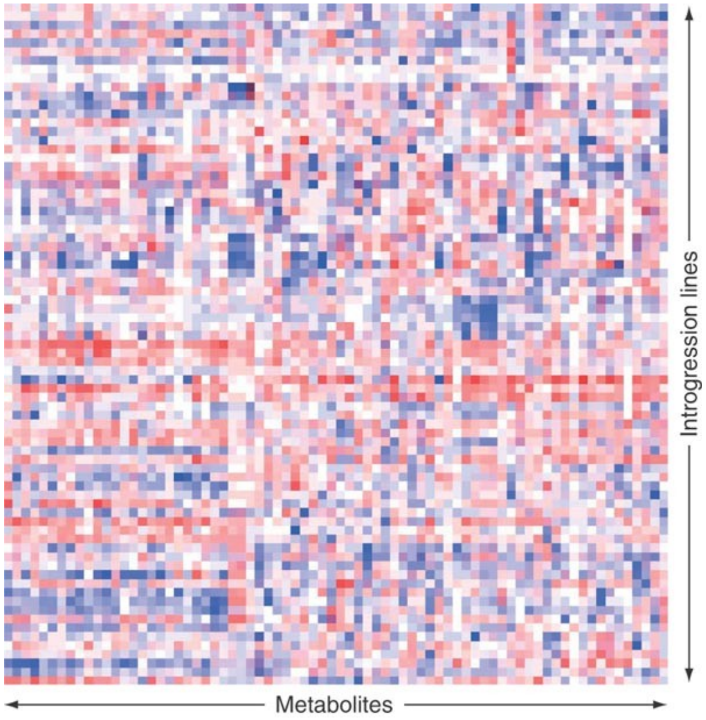
- Multivariate analysis:
  - principal component analysis (PCA)
  - visualization of high-dimensional data
  - clustering
- Least-squares linear regression
- Curve fitting
  - e.g. for time-course data using kinetic models

High-throughput biology data boom

- (c)DNA micro-arrays
- Next-generation DNA sequencing
- Untargeted Mass-Spec techniques
- Liquid handling robots
- Time-lapse microscopy



How can we “look” at the data



There is nothing better than a heat map to say:

“we gathered a lot of data ...
but we have no clue what to do with it”

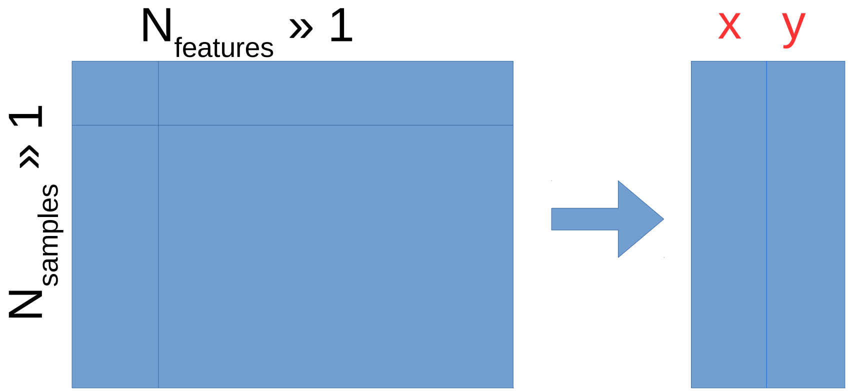
Nicolas Schauer, Yaniv Semel, Ute Roessner, Amit Gur, Ilse Balbo, Fernando Carrari, Tzili Pleban, Alicia Perez-Melis, Claudia Bruedigam, Joachim Kopka, Lothar Willmitzer, Dani Zamir & Alisdair R Fernie
Comprehensive metabolic profiling and phenotyping of interspecific introgression lines for tomato improvement
Nature Biotechnology (2006)

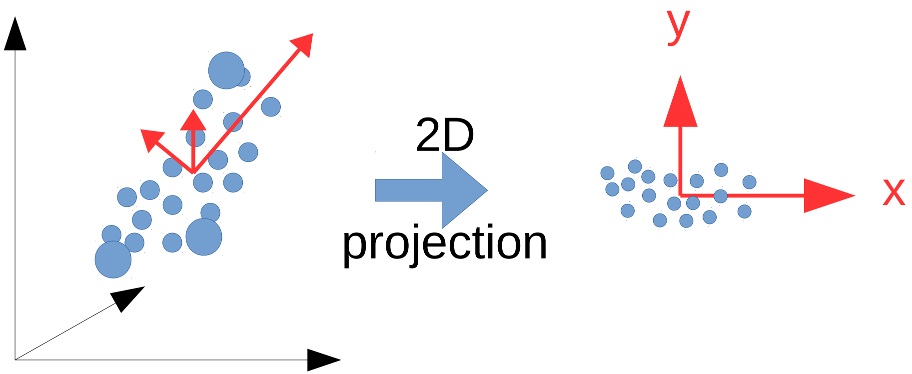
Principal Component Analysis (PCA)

- A statistical method developed in 1901 by Karl Pearson
- Commonly used to reduce the dimension of the data (e.g. 2D)

PCA implementation

- Input:
  A set of points x_1 .. x_n in high dimension ($N_{features}$)
- Output:
  A linear projection to lower dimension that best preserves Euclidean distances





PCA implementation

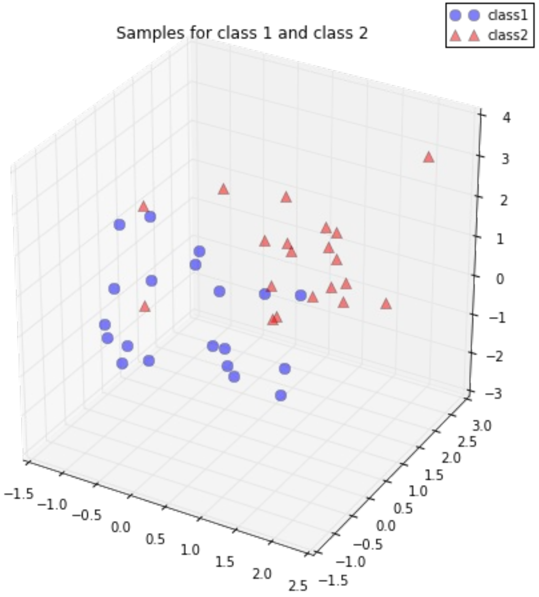
1) Arrange all samples in a ($N_{features}$ x $N_{samples}$) matrix: X

2) Subtract the mean of each feature: $\widetilde{X} = X - E(X)$

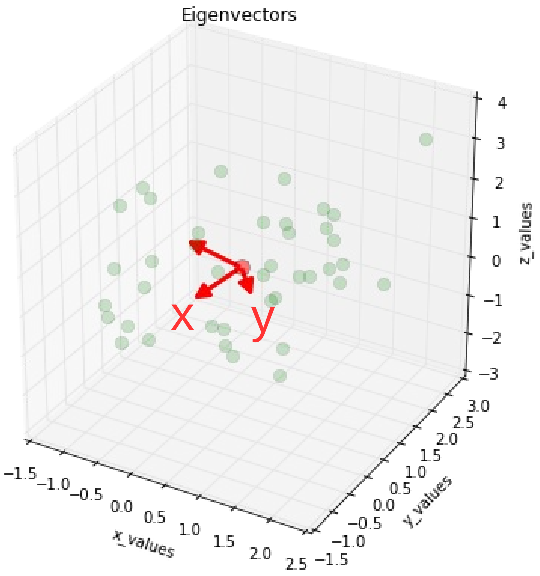
3) Calculate the Singular Value Decomposition: $U \Sigma V^T = \widetilde{X}$
make sure the eigenvalues are arranged in decreasing order

4) *U* contains the principal components

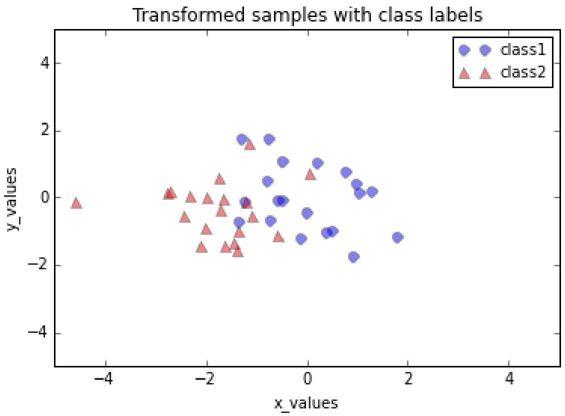
Visual example



3D data



eigenvectors



2D projection

PCA pathologies

- Assumes a multivariate Gaussian distribution
- Sensitive to relative scaling of one dimension (e.g. changing units)
- Some data cannot be easily projected into 2D without loosing much of the information (e.g. a sphere)
- Not discriminatory - treats all points as one type (doesn't see color)

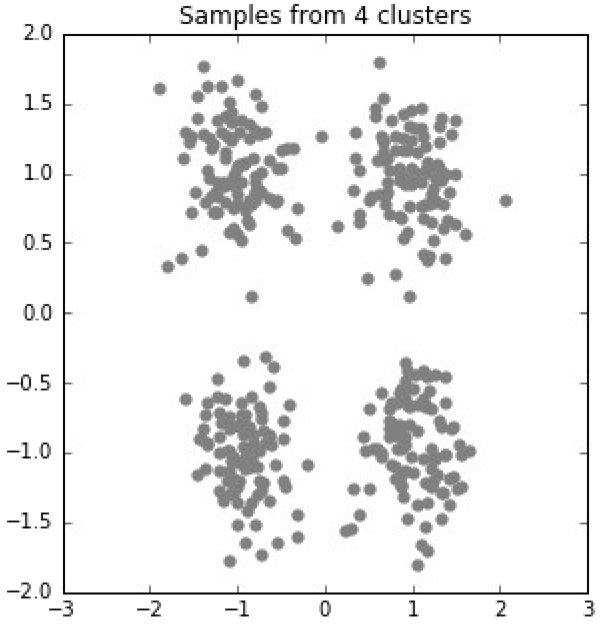
Other methods of visualization

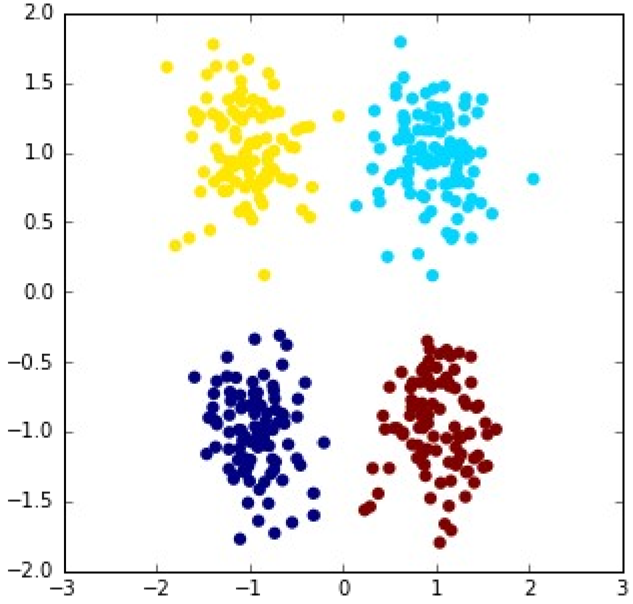
- Linear Discriminant Analysis (LDA)
- t-Distributed Stochastic Neighbor Embedding (t-SNE)

What is clustering?

- The search for “subgroups of similar objects” in a given dataset
- Objects from one subgroup should be more similar to each other than objects from other groups
- Examples:
  - finding clusters of genes with similar expression behavior over time
  - dividing of a seemingly identical disease into sub-phenotypes

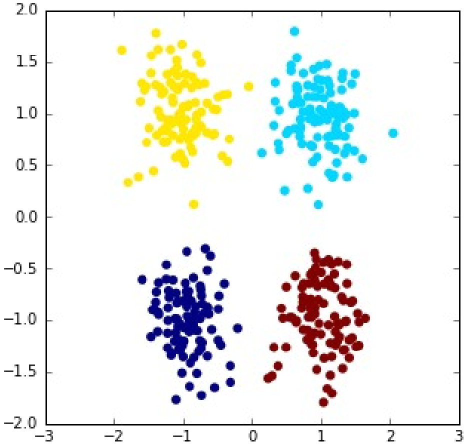
What is clustering?





Clustering: *K*-means

- <u>Input</u>:
  A set of points x_1 .. x_n and an integer $K \in \mathbb{N}$
- <u>Output</u>:
  An association of points to clusters that minimizes the within-cluster sum of squares:



$$\underset{C_k}{Minimize} \sum_{k=1}^{K} \sum_{x_n \in C_k} \left\| x_n - \mu_k \right\|^2$$

$$\mu_k \equiv \frac{1}{C_k} \sum_{x_n \in C_k} x_n$$

Lloyd's algorithm

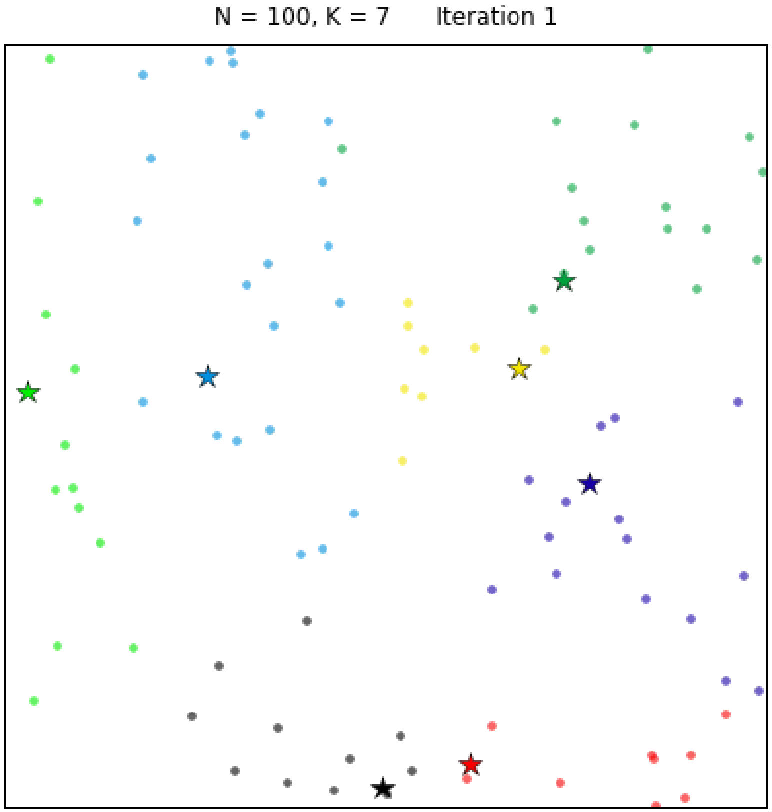
1) Randomly pick K points as initial cluster means
2) Assign each point to its nearest cluster mean:

$$\underset{k}{\arg\min} \|x_n - \mu_k\|$$

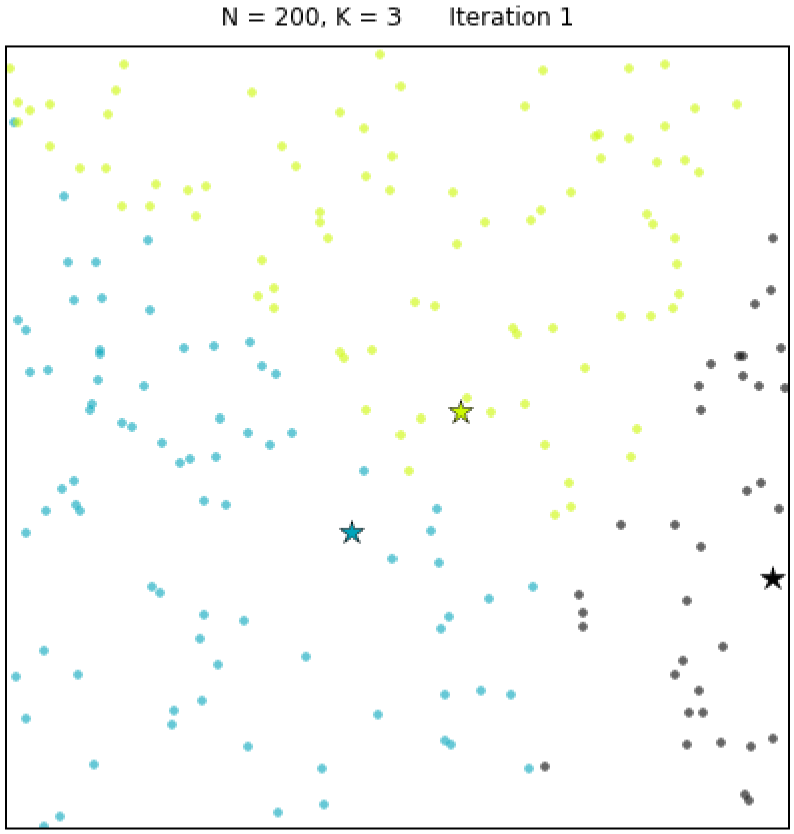
3) Recompute the mean of each cluster:

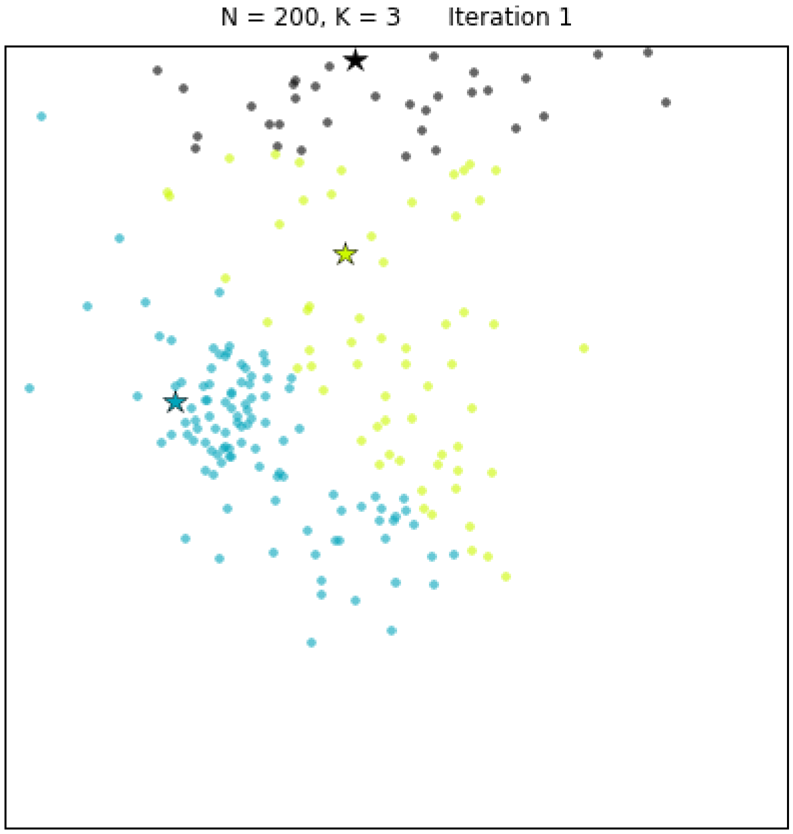
$$\mu_k = \frac{1}{C_k} \sum_{x_n \in C_k} x_n$$

4) Repeat steps 2 and 3 until cluster assignment does not change any more

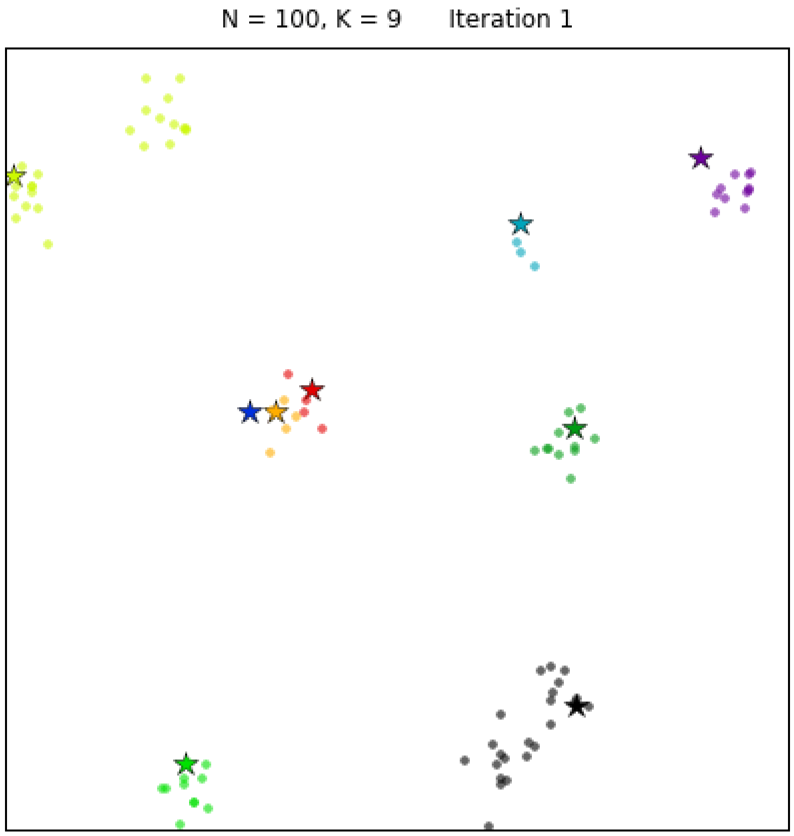


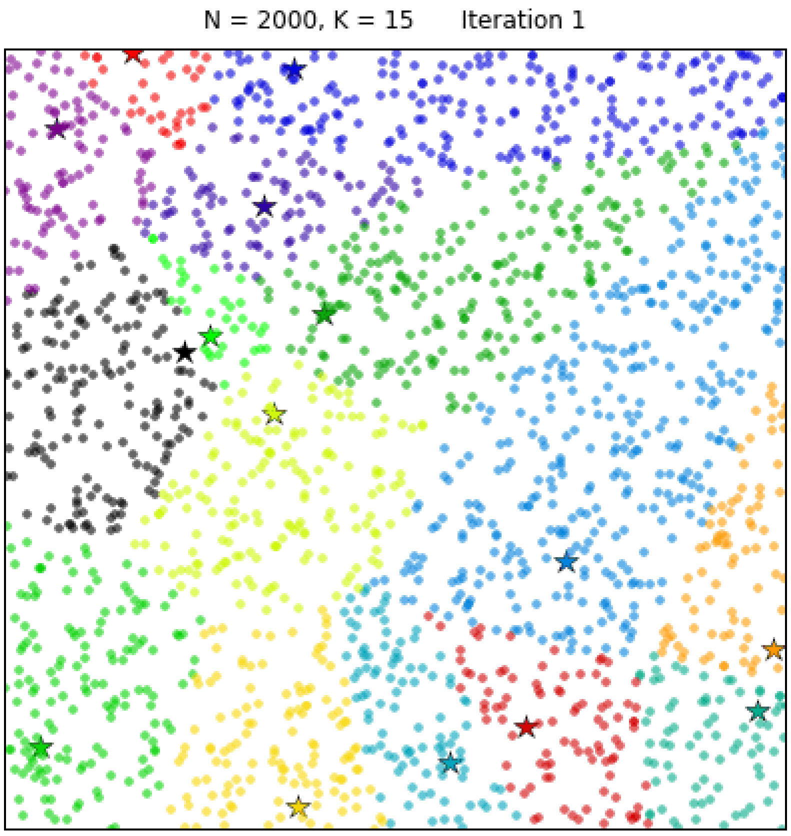
K-means examples





K-means examples





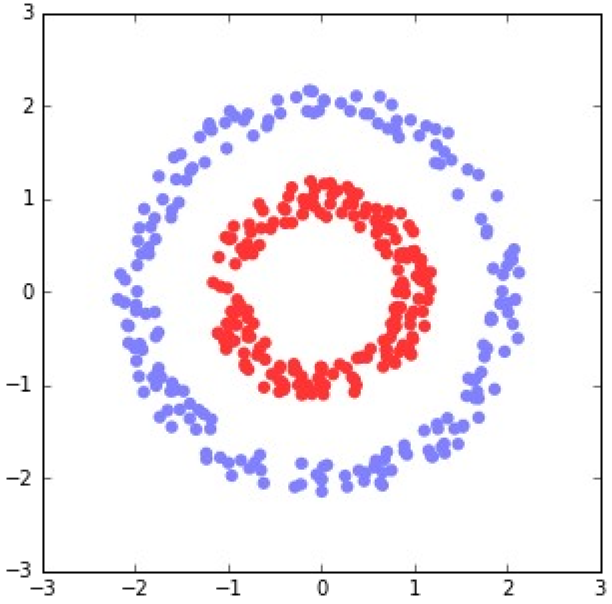
K-means pathologies

- Lloyd's algorithm can only find a local optimum, and depends on the initialization
  Solution: repeat with many randomized initial clusters

- Under-/over- estimating the number of clusters
  Solution: run for $K = 1..K_{max}$, and choose one where the average within-cluster distance drops significantly

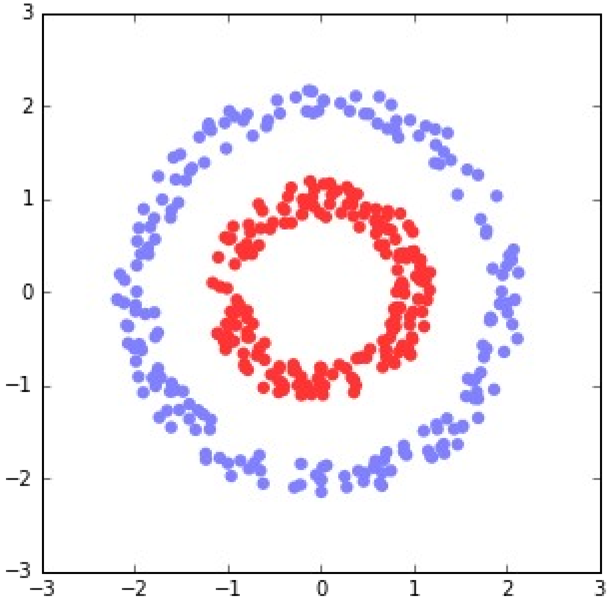
- Clusters that have non-spherical geometry
  Solution: use another method, e.g. hierarchical clustering

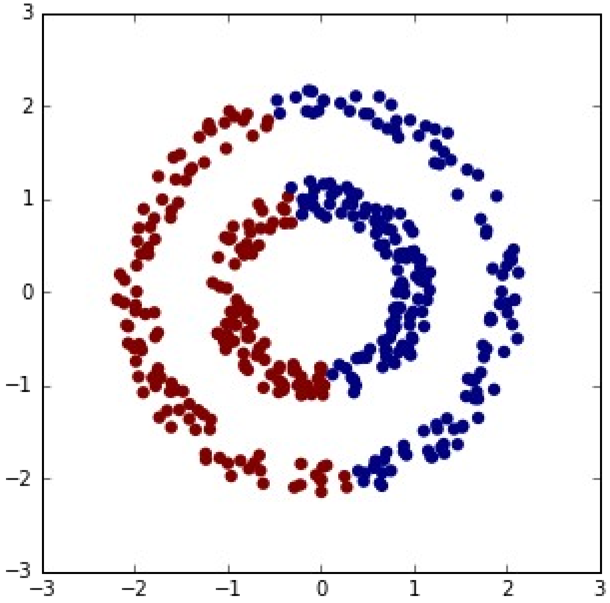
Concentric rings clustering with K-means



?

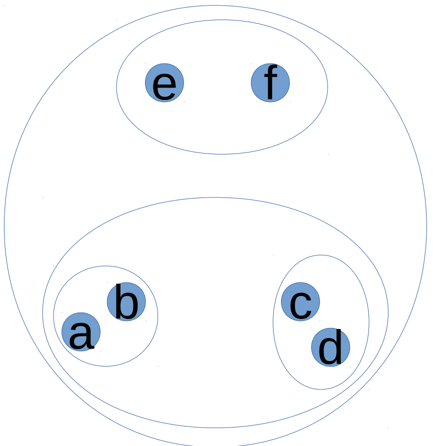
Concentric rings clustering with K-means

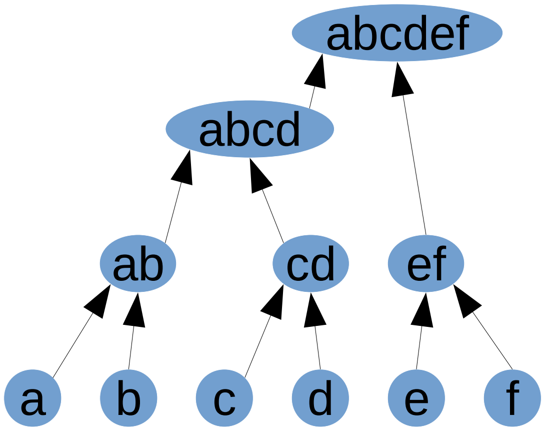


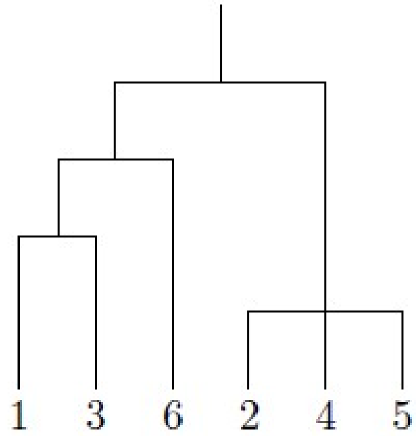


Clustering: hierarchical

- A sub-class of graph-based algorithms
- Input:
  A distance matrix *D* (size *n x n*) between each pair of data points
- Output:
  A *Dendrogram* (a tree diagram, whose leaves are the *n* points)



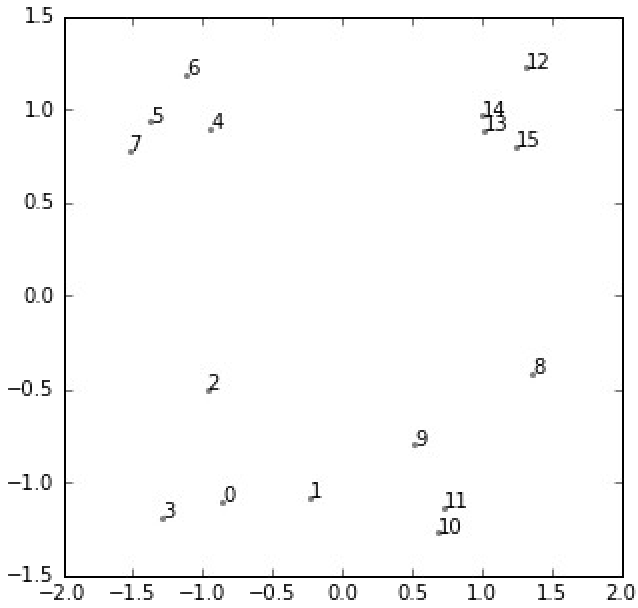




Agglomerative hierarchical clustering

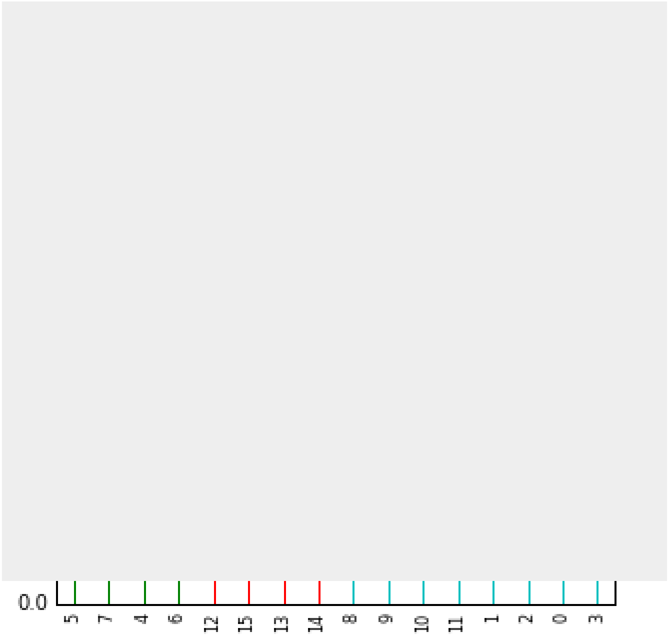
- Initialize each point to be a cluster of its own
- Repeat *n* times:
  - calculate the distance* between each two clusters
  - join the two most similar clusters

Hierarchical clustering example

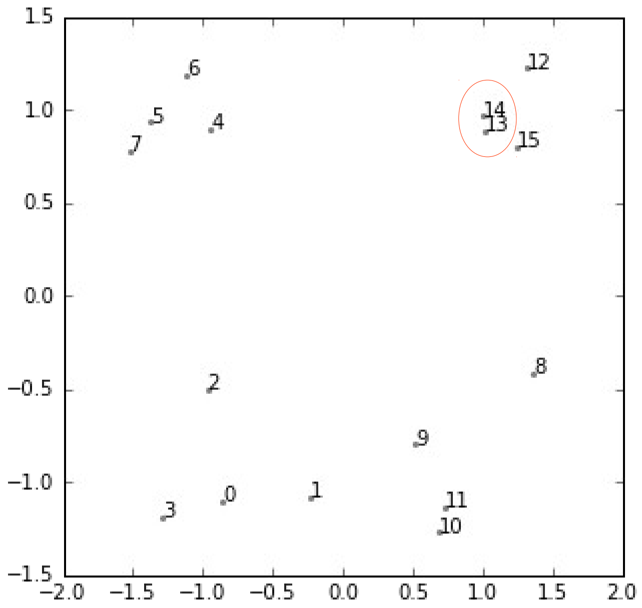


euclidean distance

single link

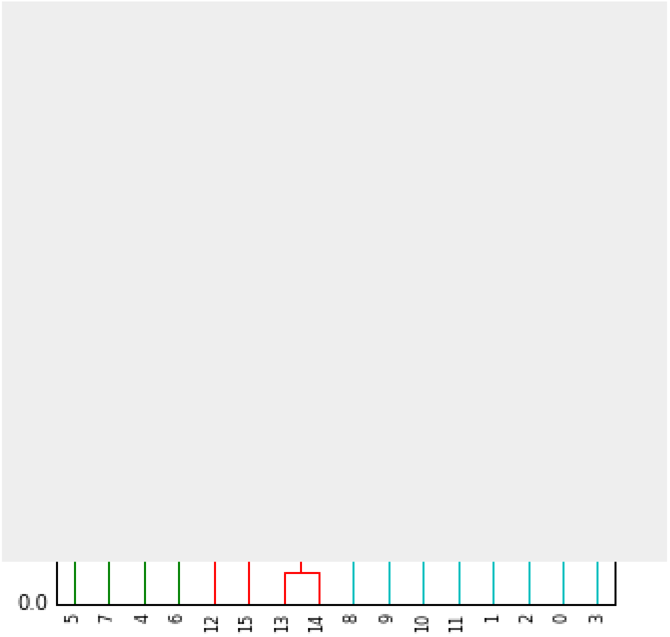


Hierarchical clustering example

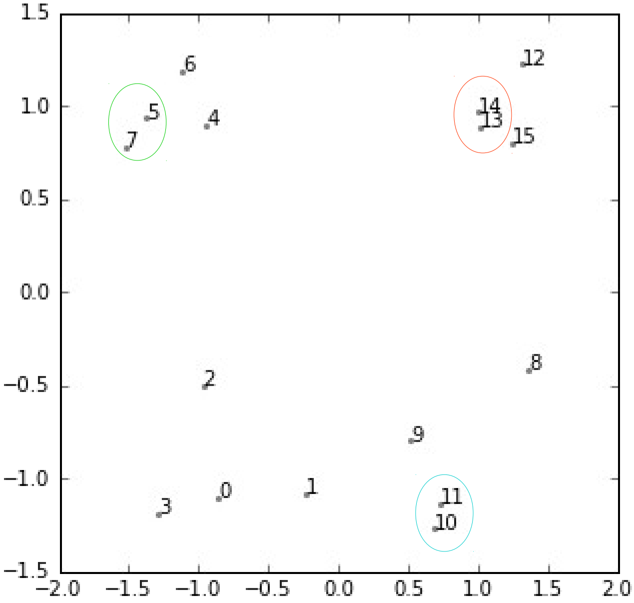


euclidean distance

single link

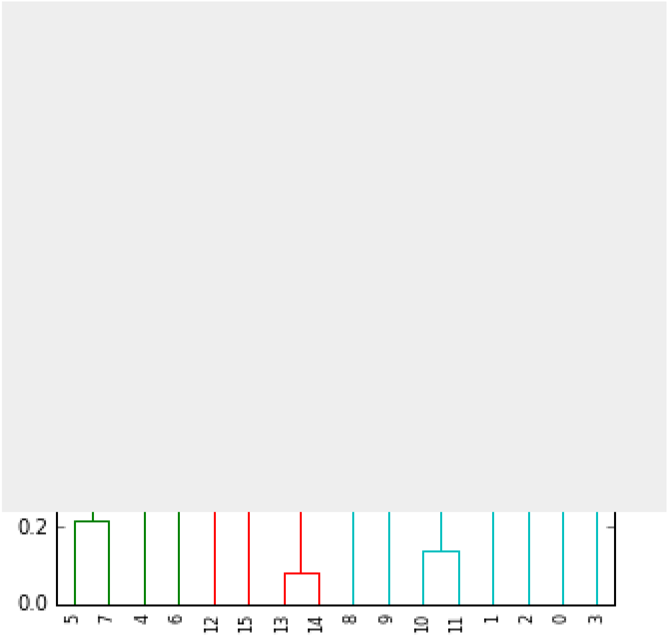


Hierarchical clustering example

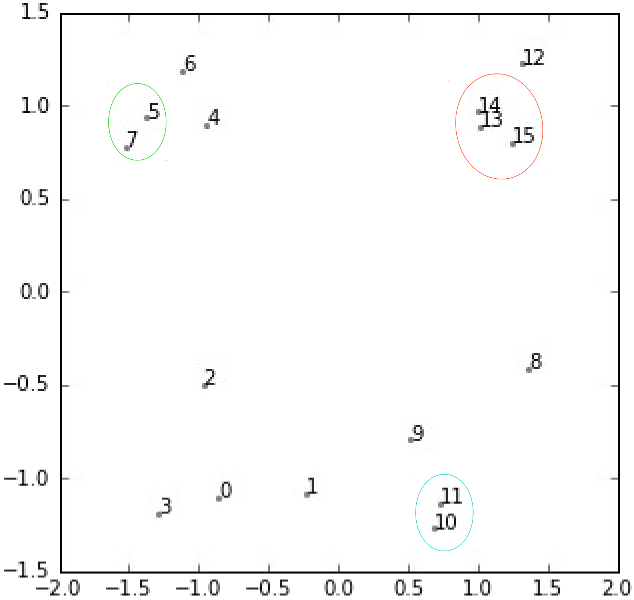


euclidean distance

single link

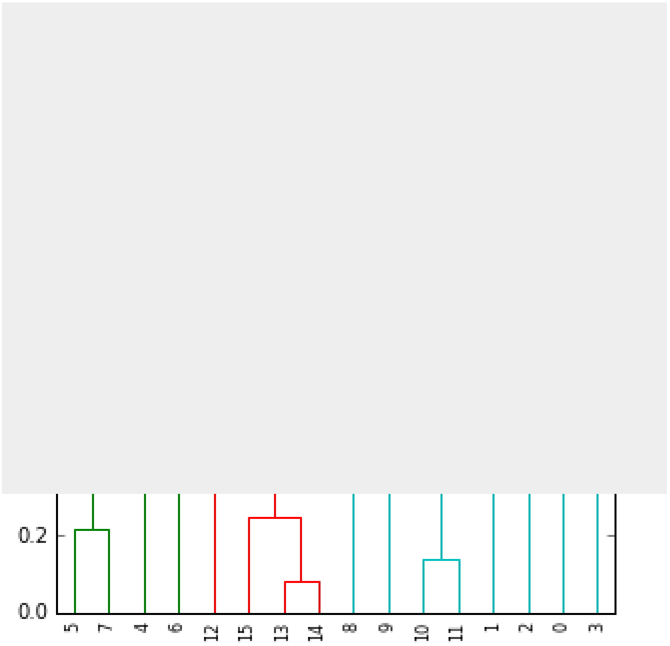


Hierarchical clustering example

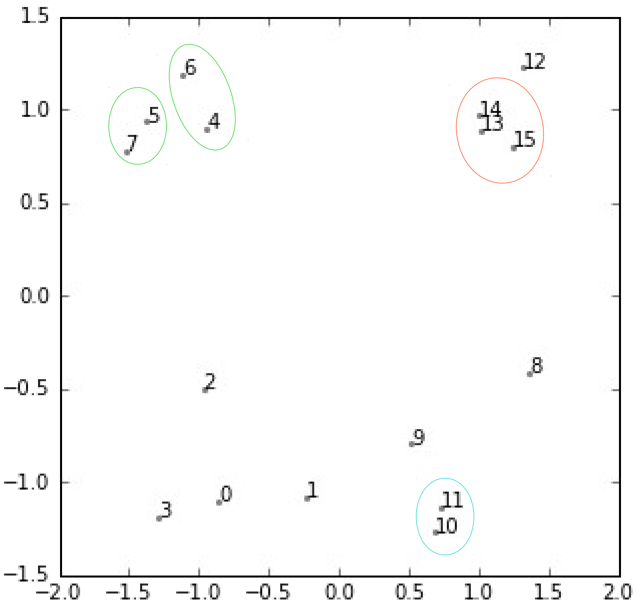


euclidean distance

single link

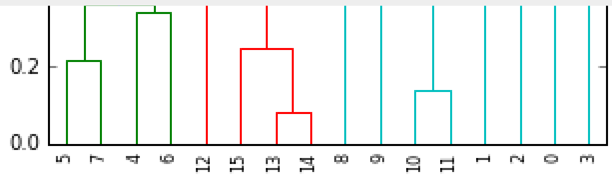


Hierarchical clustering example

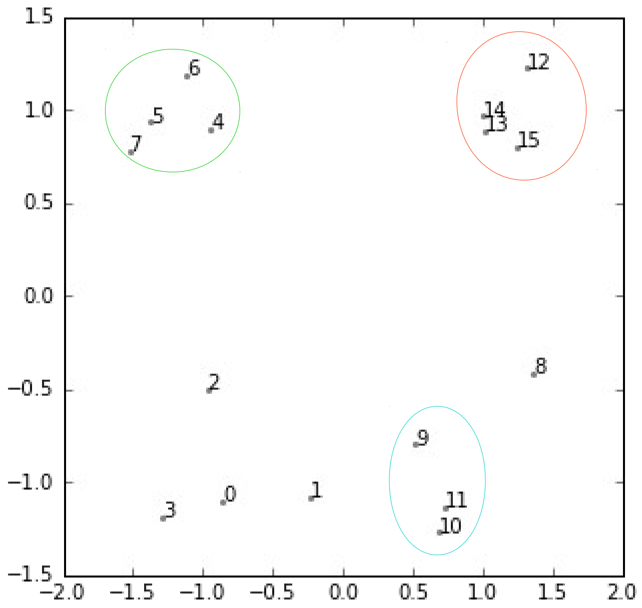


euclidean distance

single link

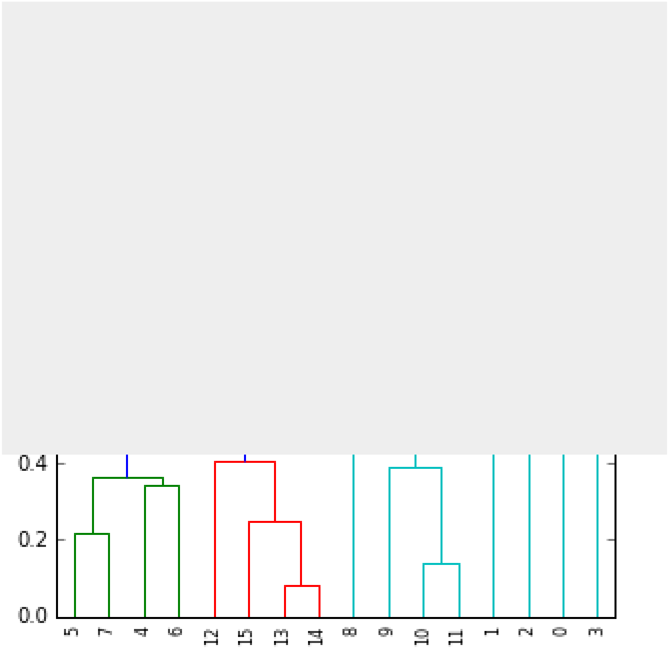


Hierarchical clustering example

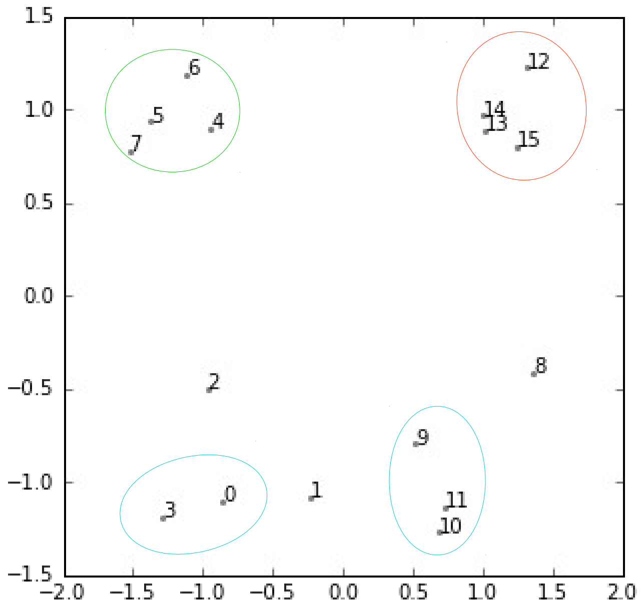


euclidean distance

single link

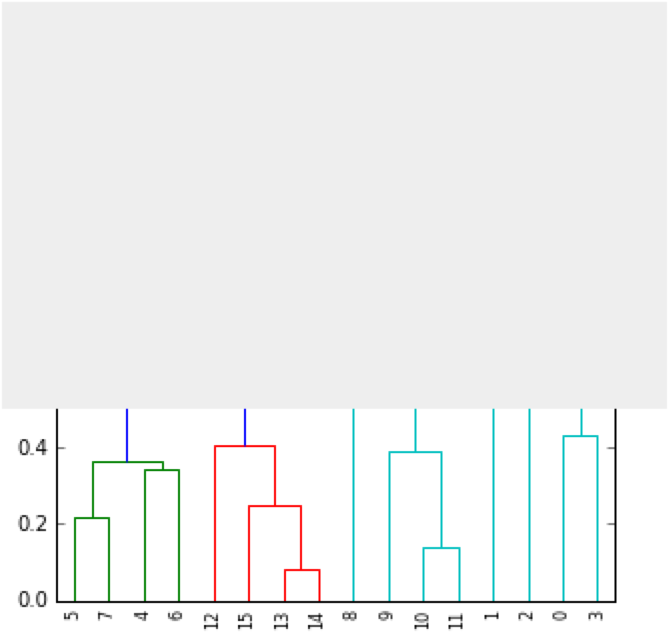


Hierarchical clustering example

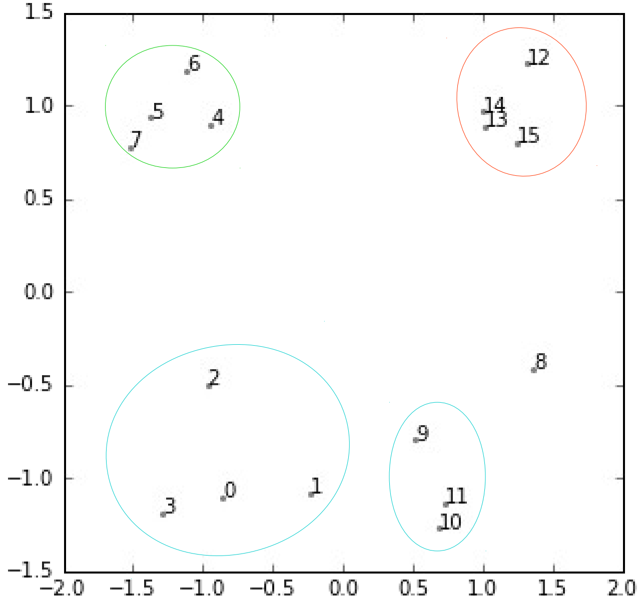


euclidean distance

single link

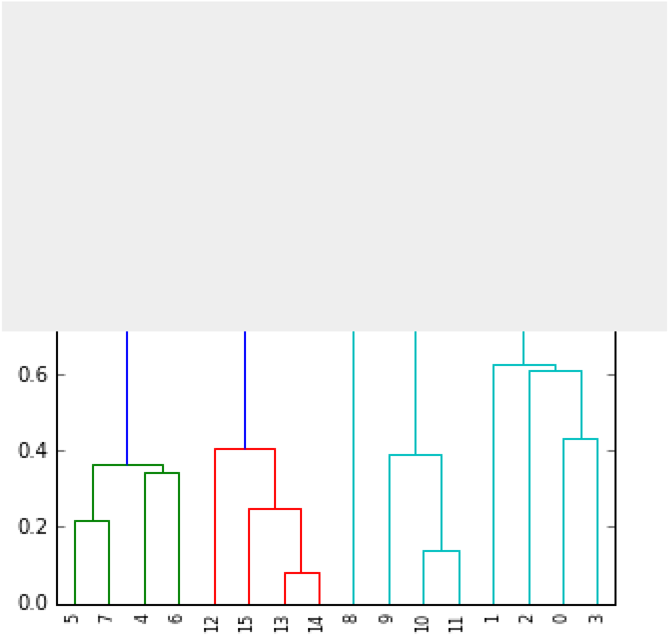


Hierarchical clustering example

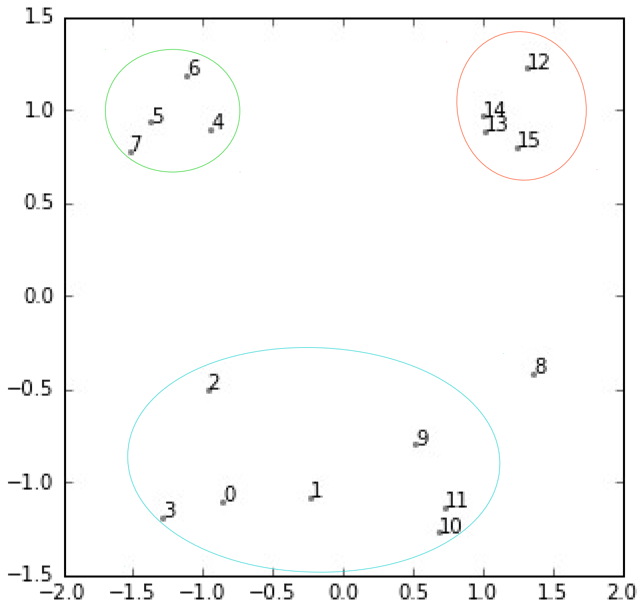


euclidean distance

single link

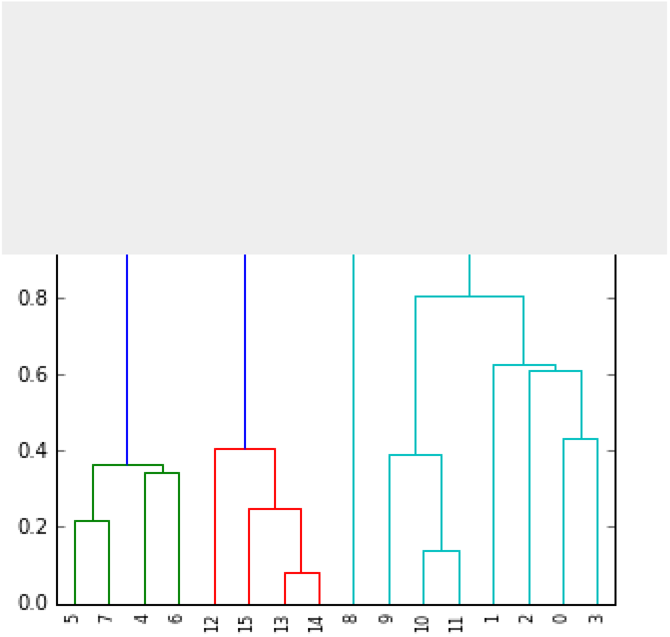


Hierarchical clustering example

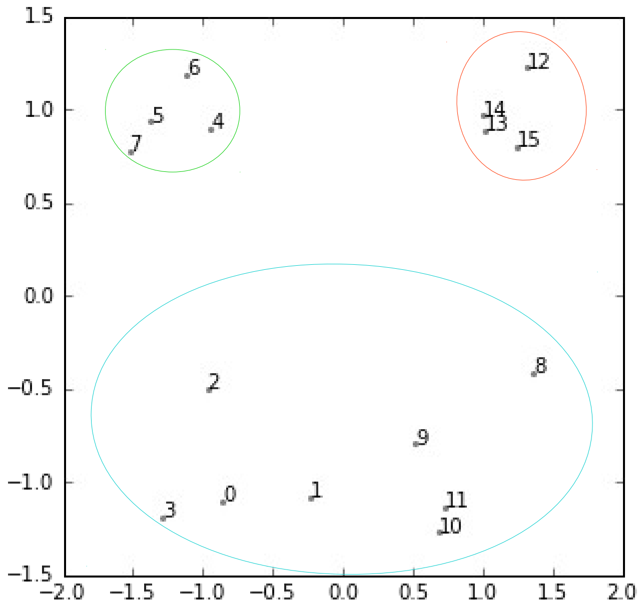


euclidean distance

single link

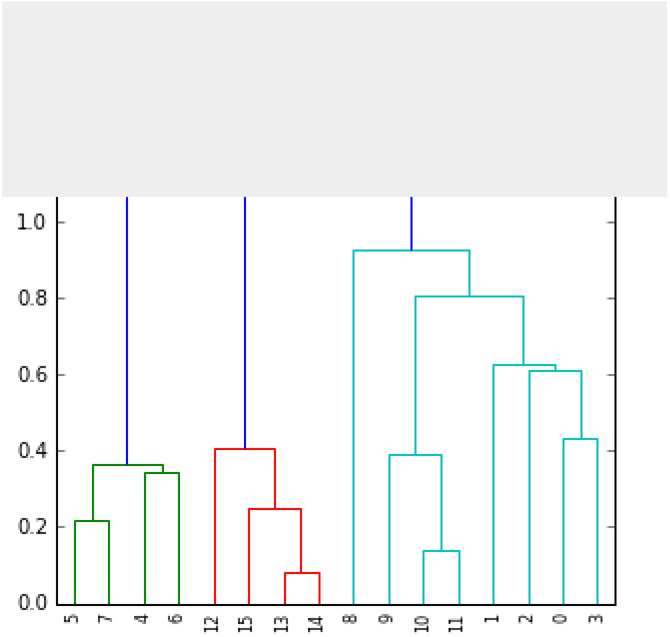


Hierarchical clustering example

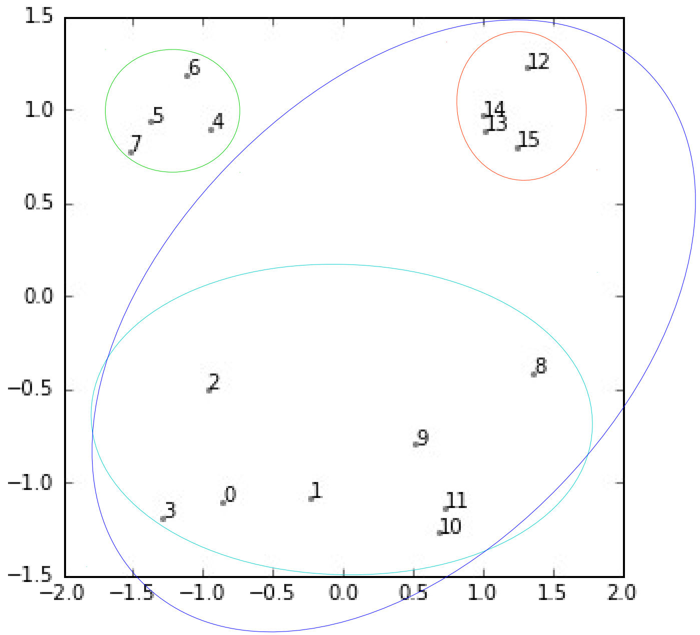


euclidean distance

single link

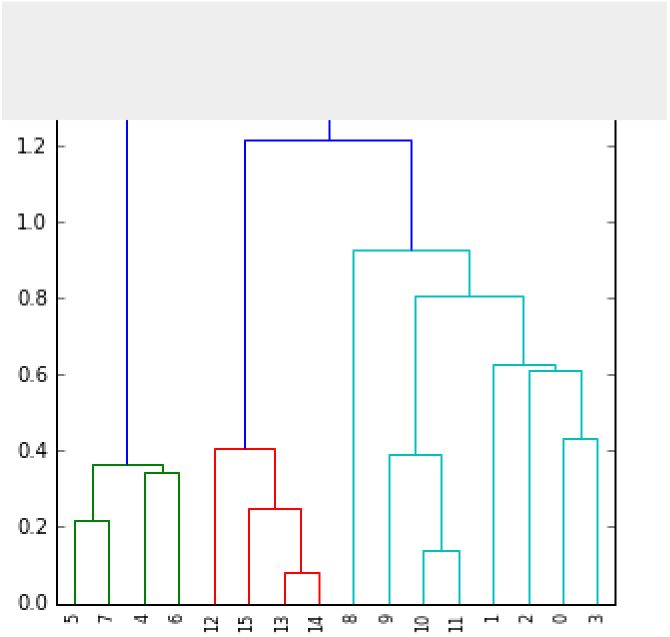


Hierarchical clustering example

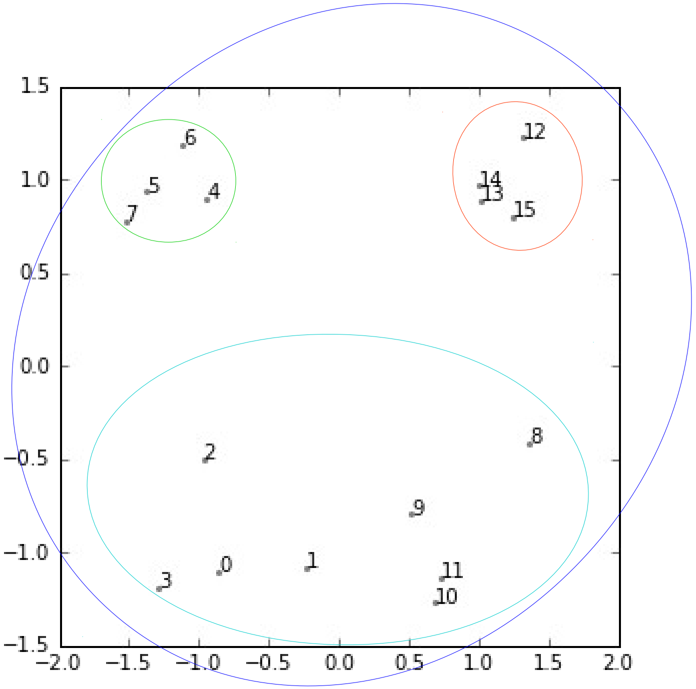


euclidean distance

single link

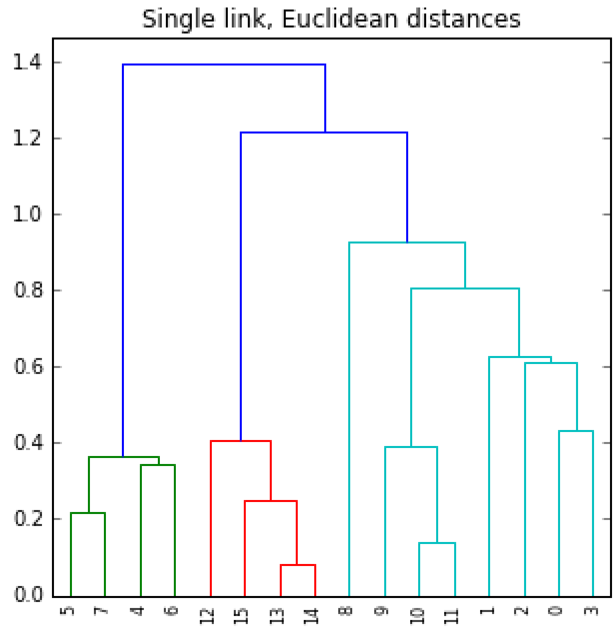


Hierarchical clustering example

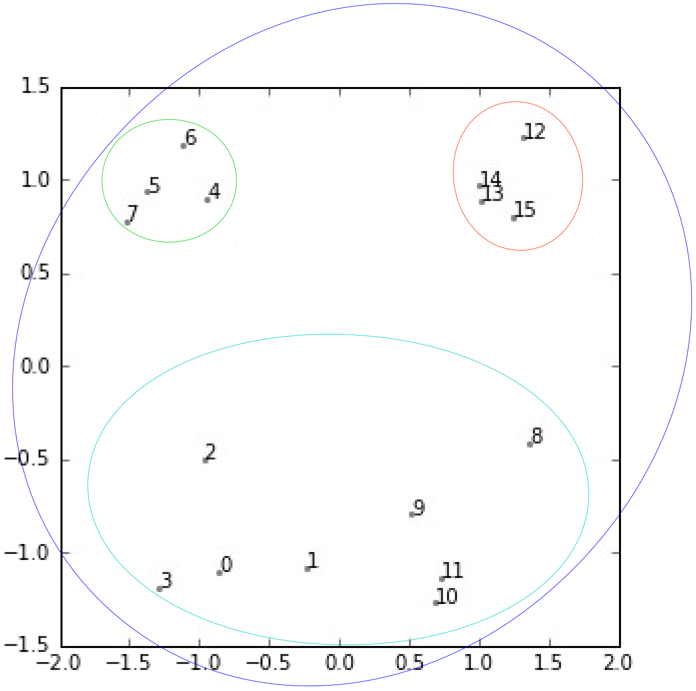


euclidean distance

single link

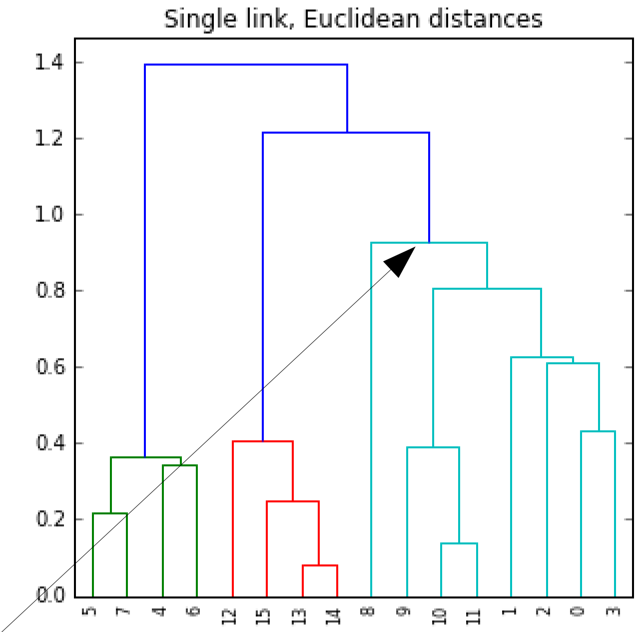


Hierarchical clustering example



euclidean distance

single link



sample 8 clusters after 0-3 and 9-11 are already in one cluster

Agglomerative hierarchical clustering

- Initialize each point to be a cluster of its own
- Repeat *n* times:
  - calculate the distance* between each two clusters
  - join the two most similar clusters

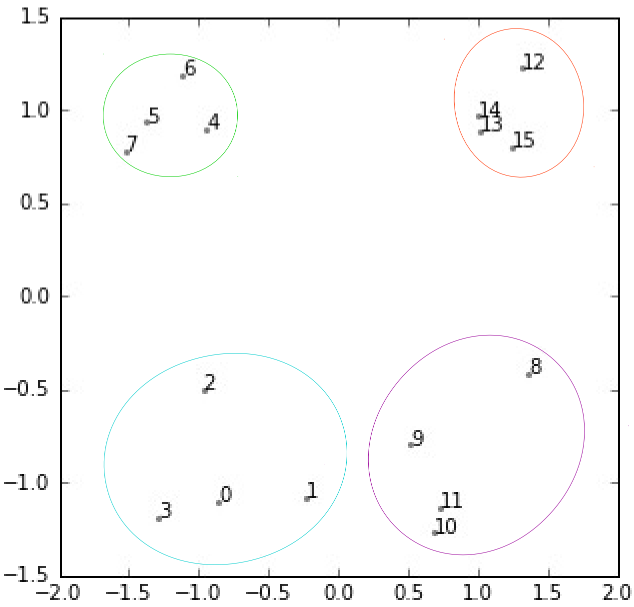
*distance between two points can be defined, for example, as:

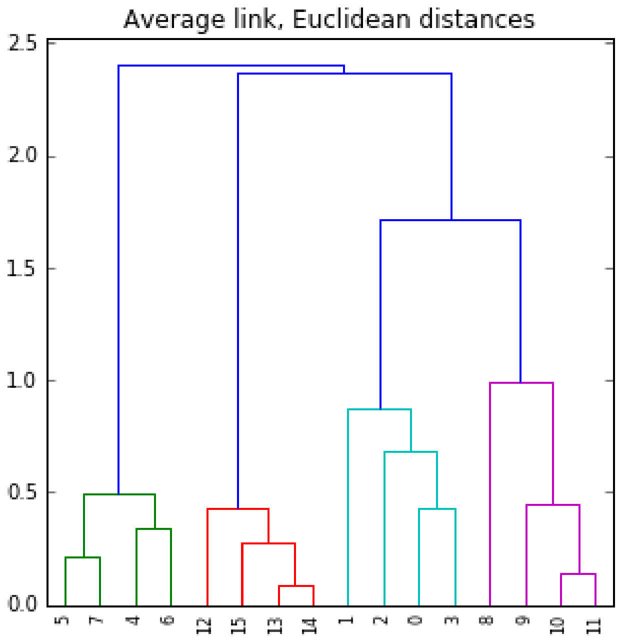
$$D_{i,j} = \|x_i - x_j\|_2 \qquad D_{i,j} = \|x_i - x_j\|_1 \qquad D_{i,j} = \|x_i - x_j\|_\infty$$

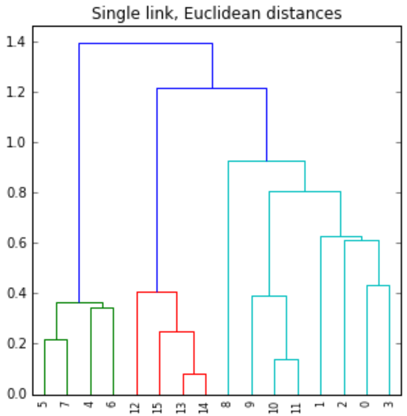
distance between two clusters can be defined, for example, as:

- complete link: $d_{complete}(A,B) = max\{D_{i,j} | x_i \in A, x_j \in B\}$
- average link: $d_{average}(A,B) = mean\{D_{i,j} | x_i \in A, x_j \in B\}$
- single link: $d_{single}(A,B) = min\{D_{i,j} | x_i \in A, x_j \in B\}$

Hierarchical clustering example







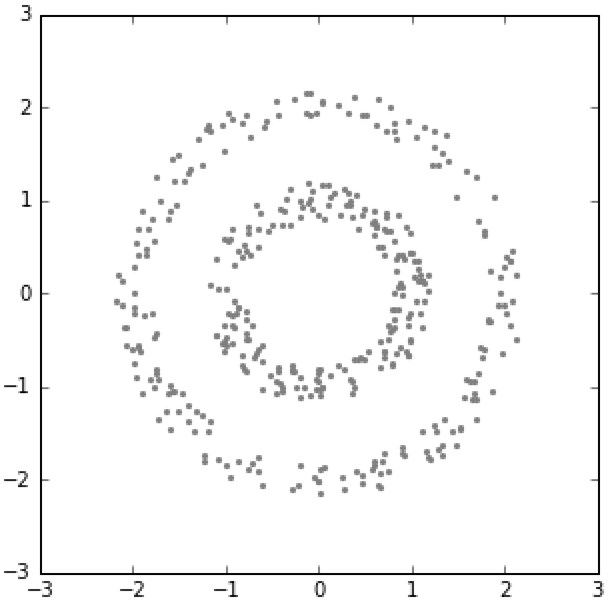
Hierarchical clustering pathologies

- Does not depend on initialization, but still there are several parameters to choose from (linkage and distance function)

- Instead of *K* parameter, one must choose a distance threshold to stop joining clusters

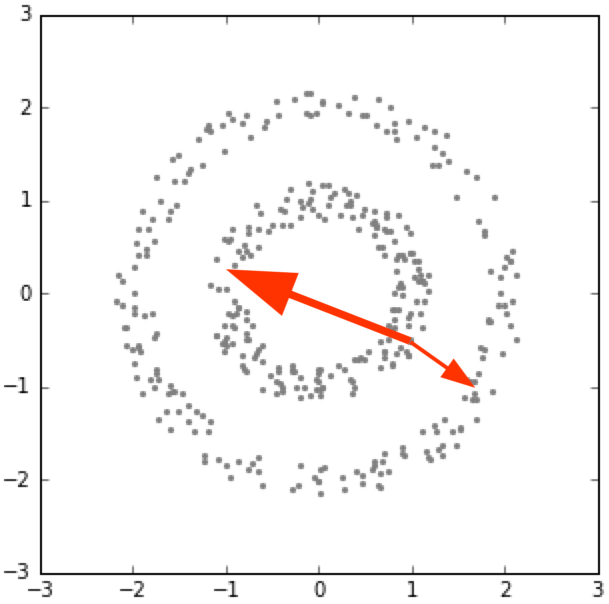
- Advantage: copes well with non-spherical clusters

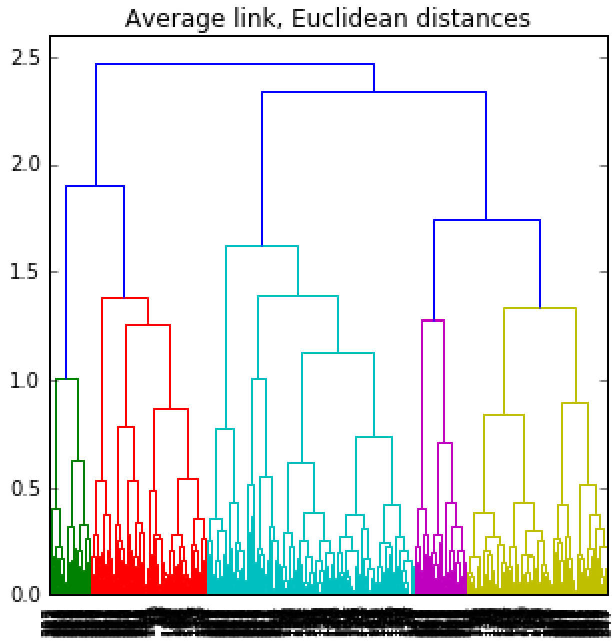
Concentric rings hierarchical clustering



?

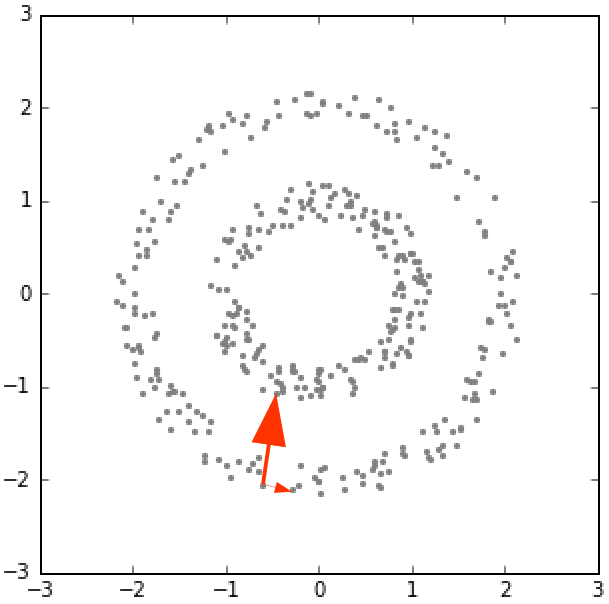
Concentric rings hierarchical clustering

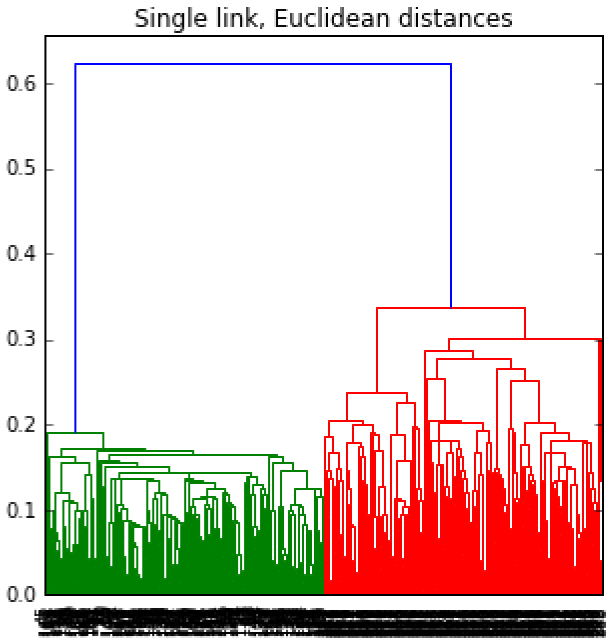




In this case, **average** link performs poorly, because some points on the other ring, are actually close than ones on the opposite side of the same ring

Concentric rings hierarchical clustering

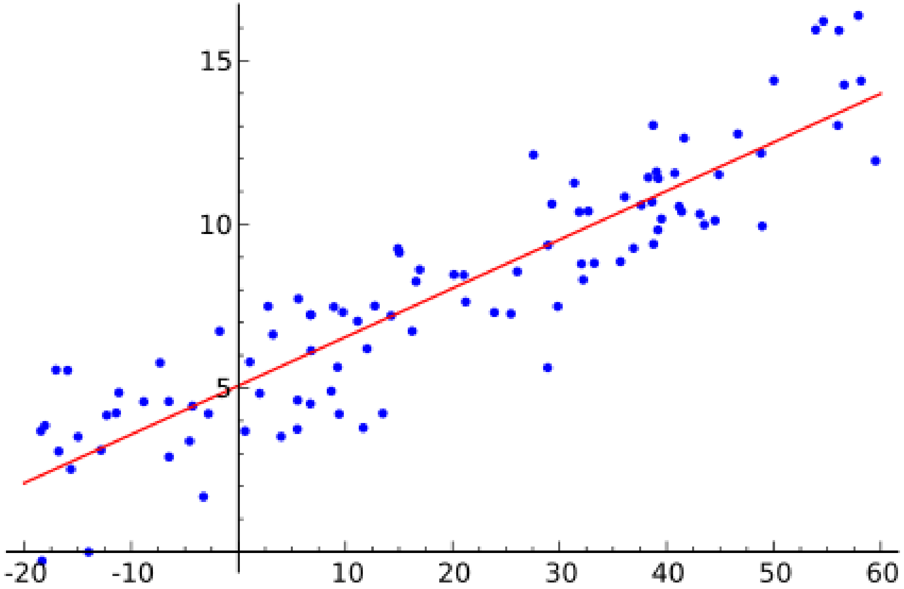




Single link, however, only looks at the minimum distance, and there is always a closer point on the same ring than the distance to the other ring

Least-squares linear regression

predicted variable

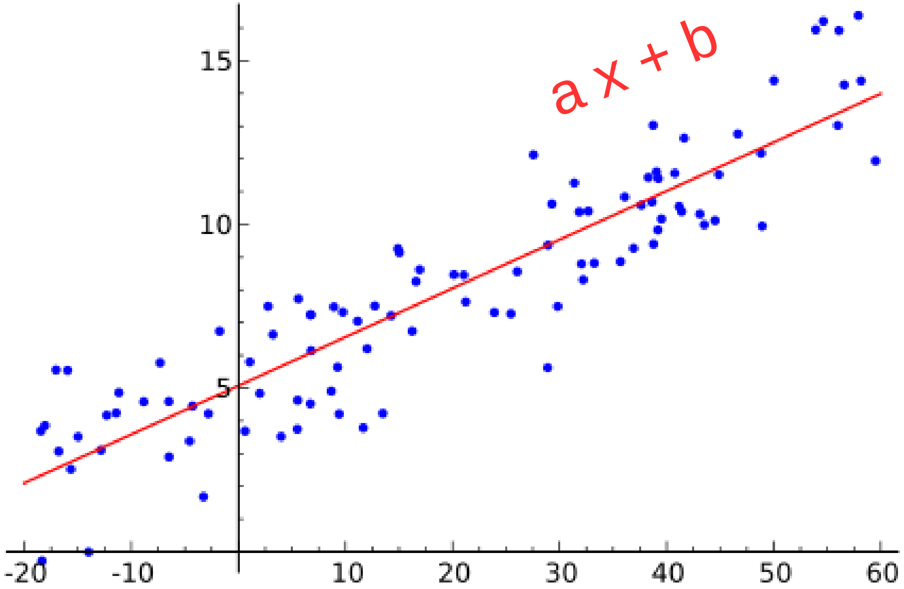


observed features

- Fits a line that minimizes distances to all the points
- Used to test for linear relationships between variables
- Usually, R^2 is used as a measure for goodness of fit
- Quite simple to implement

Ordinary Least Squares (in 2D)

predicted variable (y)



observed features (x)

$$Y \simeq a \cdot X + b$$

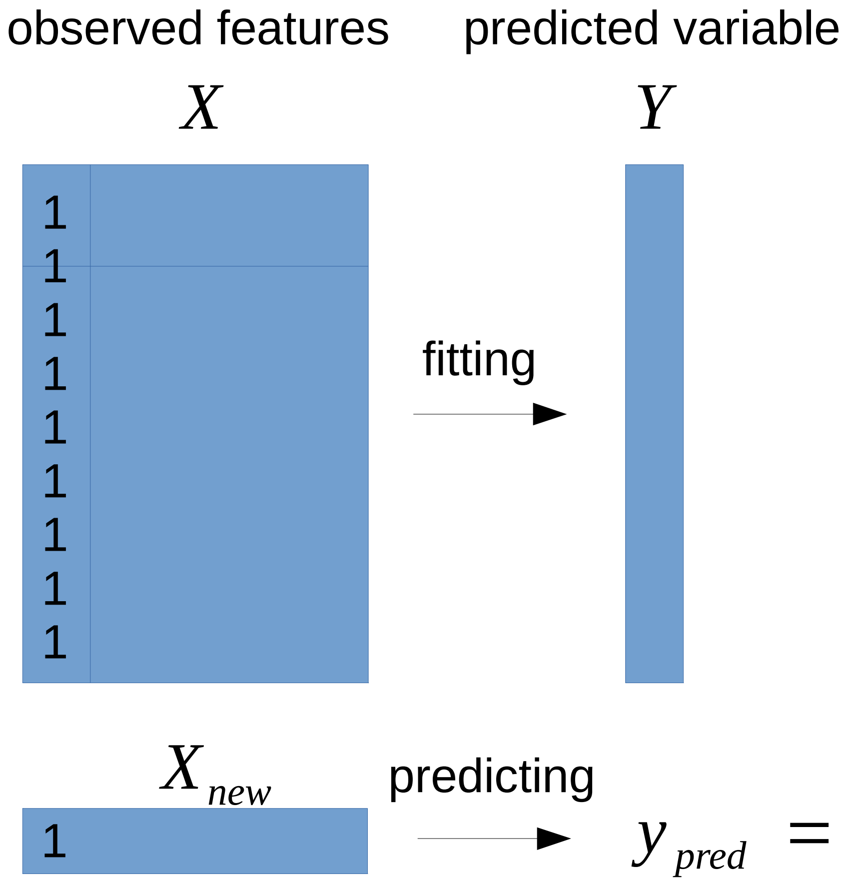
$$(a,b) = \underset{a,b}{argmin} \sum_i (y_i - a \cdot x_i + b)^2$$

solution

$$a = \frac{E(XY) - E(X)E(Y)}{E(X^2) - E(X)^2}$$

$$b = a\,E(X) - E(Y)$$

Ordinary Least Squares



model:

$$y = X\beta + \epsilon$$

$$\hat{\beta} = \underset{\beta}{argmin} \|y - X\beta\|^2$$

solution

$$\hat{\beta} \; = \; (X^T X)^{-1} X^T Y$$

$$y_{pred} \; = \; X_{new} \hat{\beta}$$

Quantifying the goodness of fit

Coefficient of determination (*R*):

$$R^2 = \frac{Var\left(Y - X\hat{\beta}\right)}{Var\left(Y\right)}$$

Pearson's correlation coefficient (*r*):

$$r^2 = \frac{cov\left(X, Y\right)}{\sigma_X \sigma_Y}$$

For ordinary linear regression in 2D:

$$r^2 = R^2$$

Quantifying the goodness of fit

Coefficient of determination (*R*):

$$R^2 = \frac{Var\left(Y - X\hat{\beta}\right)}{Var\left(Y\right)}$$

Pearson's correlation coefficient (*r*):

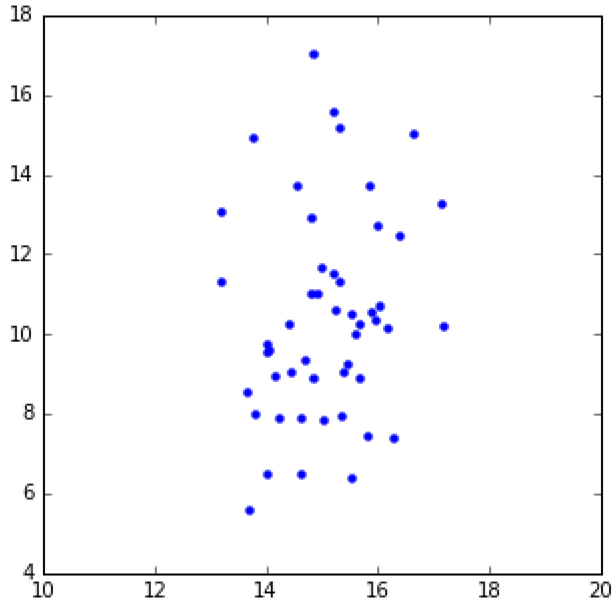
$$r^2 = \frac{cov\left(X, Y\right)}{\sigma_X \sigma_Y}$$

For ordinary linear regression in 2D:

$$r^2 = R^2$$

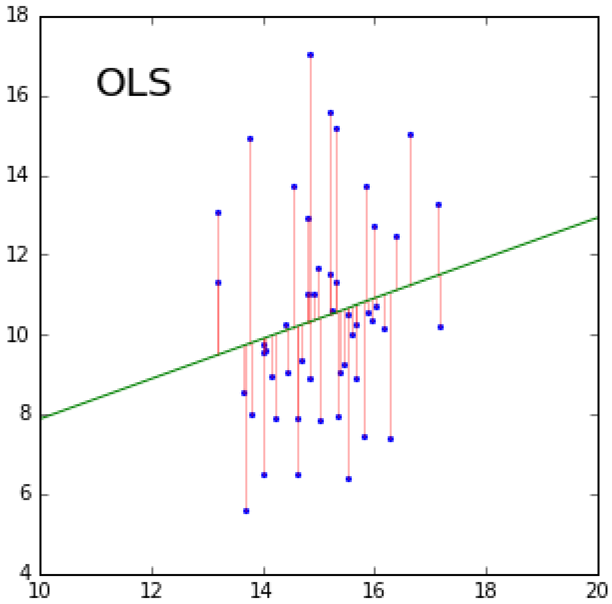
Total Least Squares

- Ordinary least-squares minimizes only the y-axis residuals
- This fits well in situations where *X* are observed with high precision, and only the *Y* values have errors (ε)



Total Least Squares

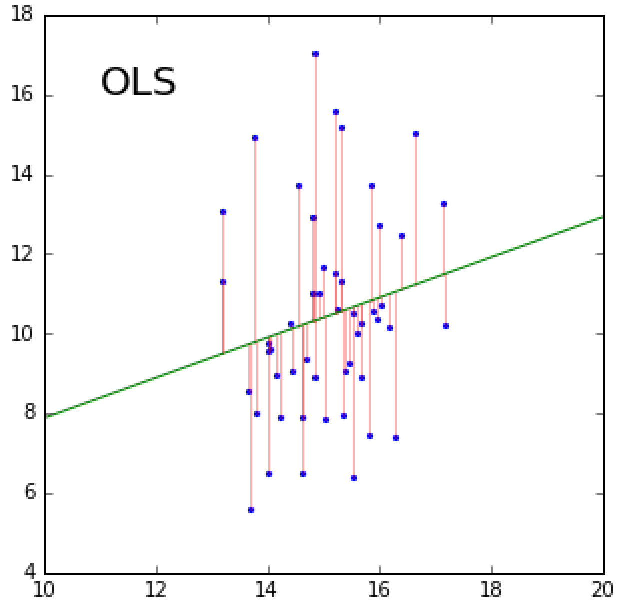
- Ordinary least-squares minimizes only the y-axis residuals
- This fits well in situations where *X* are observed with high precision, and only the *Y* values have errors (ε)
- However, when both X and Y are error prone, this doesn't always work well

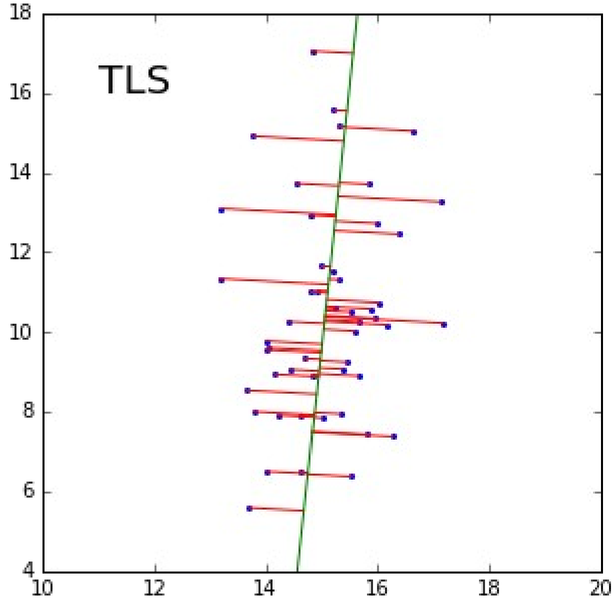


This line minimizes the y-axis residuals and ignores the x-axis ones

Total Least Squares

- <u>Solution</u>: using the total least squares algorithm (AKA orthogonal least-squares)





Total Least Squares

OLS

$$y = X\beta + \epsilon$$

$$\hat{\beta} = \underset{\beta}{argmin}\, ||\epsilon|| = \underset{\beta}{argmin} \left\| y - X\beta \right\|$$

<u>TLS</u>:

- combine X and Y into one variable Z and center it (i.e. E[Z] = 0)
- find a 1D orthogonal projection (P) minimizing the sum of residuals

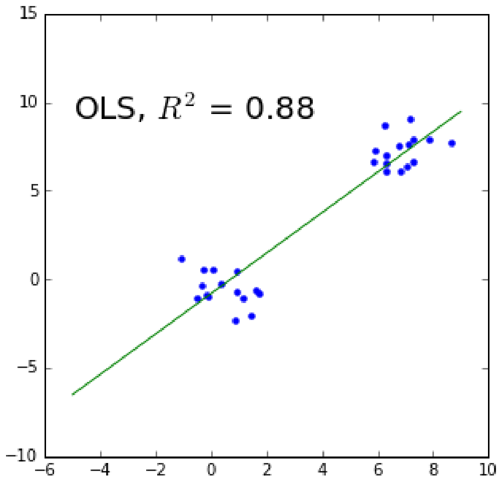
$$P = \beta\beta^T \qquad \epsilon = Z - ZP$$

$$\hat{\beta} = \underset{\beta}{argmin}\, ||\epsilon|| = \underset{\beta}{argmin} \left\| Z(I - \beta\beta^T) \right\|$$

this is the same as PCA, i.e. taking the β the component with the largest eigenvalue

Things to remember before regressing

- If the x-value have significant errors, use TLS
- Use a “natural” scale (e.g. log-scale is often required)
- Always report *N* along with the R^2
- If the distribution of points is very skewed (e.g. two distant clusters) R^2 might be misleading



General least-squares curve-fitting

- In general, curve fitting is performed by iterative minimization of the residuals
- Functions with more parameters will fit better, but will take longer to optimize and might result in over-fitting
-

CONCLUDING REMARKS