Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Unsupervised Learning

Borchert, Dr. Schapranow

Data Management for Digital Health
Winter 2023

Agenda
Pillars of the Lecture

Hasso Plattner Institut

Medical Use Cases

Biology Recap

Oncology

Nephrology

Infectious Diseases

Technology Foundation

Data Sources

Data Formats

Processing and Analysis

Software Architectures

Machine Learning

Data

ML

Refine

Evaluate

Prediction + Probability

Unsupervised Learning

Agenda
Pillars of the Lecture

Hasso Plattner Institut

Medical Use Cases


Biology Recap


Oncology


Nephrology

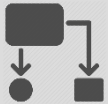

Infectious Diseases

Technology Foundation

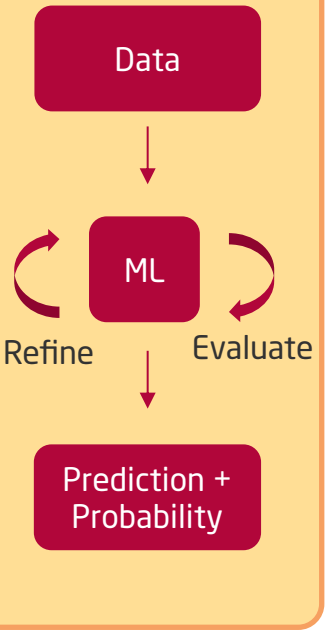

Data Sources


Data Formats


Processing and Analysis

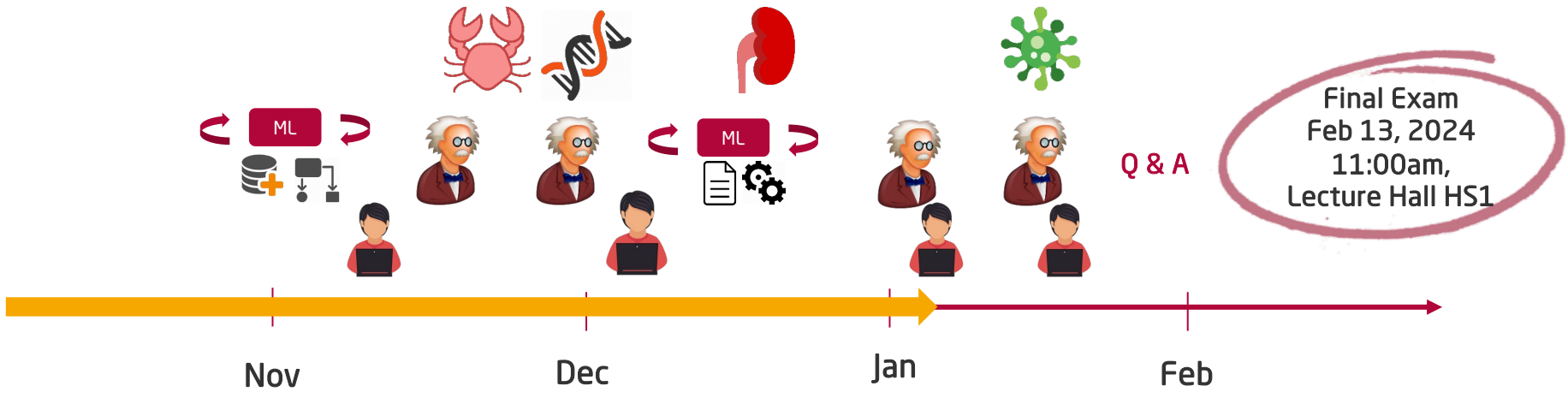

Software Architectures

Machine Learning



Unsupervised Learning

Lecture Schedule



- Lecture Kickoff
- Actors in Healthcare
- Digital Health Data

- Machine Learning (ML) Foundations
- Use Case Oncology
- Biology Recap

- Natural Language Processing
- Use Case Nephrology & Intensive Care
- Supervised ML & Deep Learning

- Use Case Infectious Diseases
- Unsupervised ML

Agenda

- Similarity Measures
- Clustering Algorithms
  - K-Means Clustering
  - Gaussian Mixtures
  - DBSCAN
  - Agglomerative Hierarchical Clustering
- Evaluation of Clustering Results

Problem Settings in Machine Learning

Hasso Plattner Institut

Supervised Learning (Labels available for training)

Classification

Categorical output

e.g. x ∈ Fruits, y ∈ {"apple", "orange"}

f()= "apple"

f()= "orange"

Regression

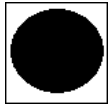
Continuous output

e.g.: x ∈ Fruits, y ∈ $\mathbb{R}_+$ ≙ t until ripe)

f () = 12 days

Structured Prediction

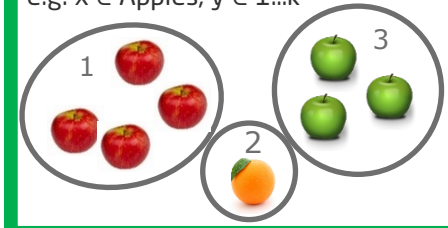
e.g. x ∈ $\mathbb{R}^{w \times h \times d}$, y ∈ $\mathbb{R}^{w \times h}$ ≙ pixels

f() = 

Unsupervised Learning (No labels during training)

Clustering

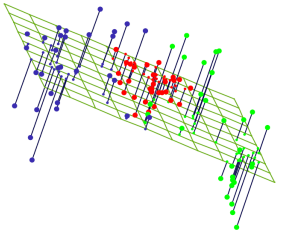
e.g. x ∈ Apples, y ∈ 1...k



Dimensionality reduction

x ∈ $\mathbb{R}^d$, x' ∈ $\mathbb{R}^p$, p < d

e.g., projecting all features of a fruit to 2 dimensions for visualization



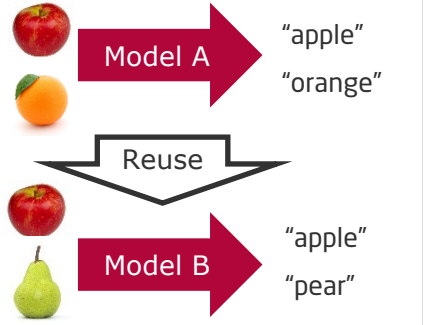
Semi-Supervised Learning (Some labels for training)

Anomaly / novelty detection

trained only on "normal" samples

e.g. x ∈ Apples, y ∈ { ☺ , ☹ }

Transfer Learning



Reinforcement Learning

https://en.wikipedia.org/wiki/Apple
https://cdn4.vectorstock.com/i/1000x1000/16/58/robot-arm-line-icon-sign-on-vector-17841658.jpg

Unsupervised Learning

Data Management for Digital Health, Winter 2023

Problem Settings in Machine Learning

Hasso Plattner Institut

Supervised Learning (Labels available for training)

Classification

Categorical output

e.g. x ∈ Fruits, y ∈ {"apple", "orange"}

f( )= "apple"

f( )= "orange"

Regression

Continuous output

e.g.: x ∈ Fruits, y ∈ $\mathbb{R}_+$ ≙ t until ripe)

f ( ) = 12 days

Structured Prediction

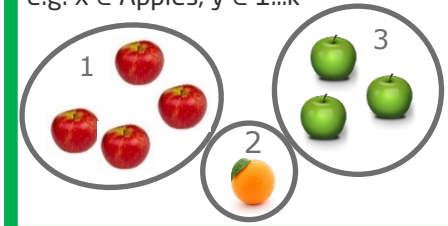
e.g. x ∈ $\mathbb{R}^{w \times h \times d}$, y ∈ $\mathbb{R}^{w \times h}$ ≙ pixels

f( ) =

Unsupervised Learning (No labels during training)

Clustering

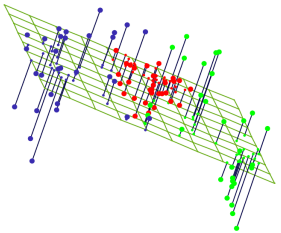
e.g. x ∈ Apples, y ∈ 1...k



Dimensionality reduction

$x \in \mathbb{R}^d$, $x' \in \mathbb{R}^p$, $p < d$

e.g., projecting all features of a fruit to 2 dimensions for visualization

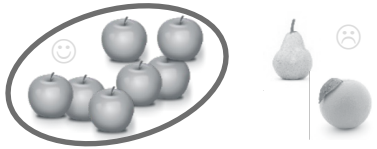


Semi-Supervised Learning (Some labels for training)

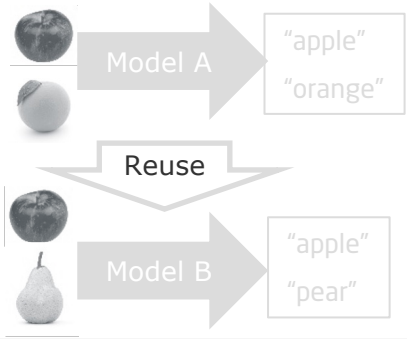
Anomaly / novelty detection

trained only on "normal" samples

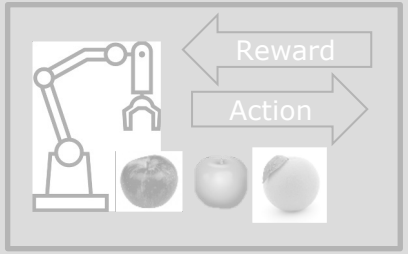
e.g. x ∈ Apples, y ∈ { ☺ , ☹ }



Transfer Learning



Reinforcement Learning



https://en.wikipedia.org/wiki/Apple
https://cdn4.vectorstock.com/i/1000x1000/16/58/robot-arm-line-icon-sign-on-vector-17841658.jpg

Unsupervised Learning

Data Management for Digital Health, Winter 2023

Problem Settings in Machine Learning

Hasso Plattner Institut

Supervised Learning (Labels available for training)

Classification

Categorical output

e.g. x ∈ Fruits, y ∈ {"apple", "orange"}

f( )= "apple"

f( )= "orange"

Regression

Continuous output

e.g.: x ∈ Fruits, y ∈ $\mathbb{R}_+$ ≙ t until ripe)

f ( ) = 12 days

Structured Prediction

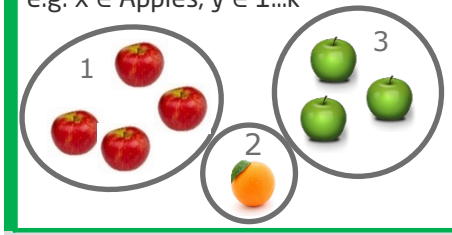
e.g. x ∈ $\mathbb{R}^{w \times h \times d}$, y ∈ $\mathbb{R}^{w \times h}$ ≙ pixels

f( ) =

Unsupervised Learning (No labels during training)

Clustering

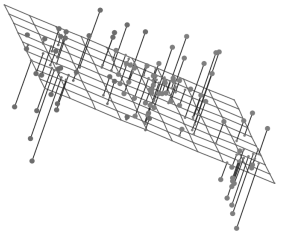
e.g. x ∈ Apples, y ∈ 1...k



Dimensionality reduction

$x \in \mathbb{R}^d$, $x' \in \mathbb{R}^p$, $p < d$

e.g., projecting all features of a fruit to 2 dimensions for visualization

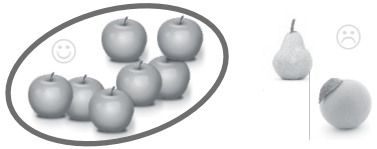


Semi-Supervised Learning (Some labels for training)

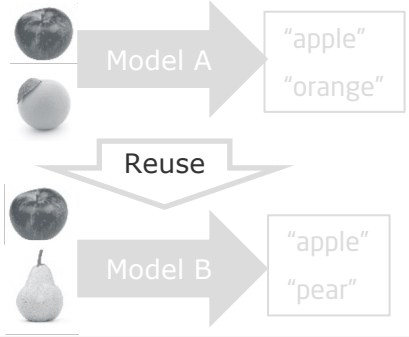
Anomaly / novelty detection

trained only on "normal" samples

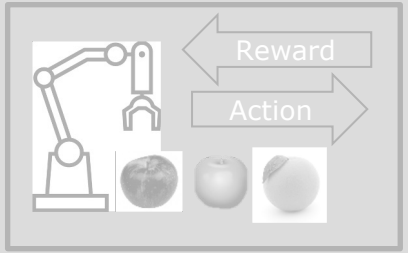
e.g. x ∈ Apples, y ∈ { ☺ , ☹ }



Transfer Learning



Reinforcement Learning

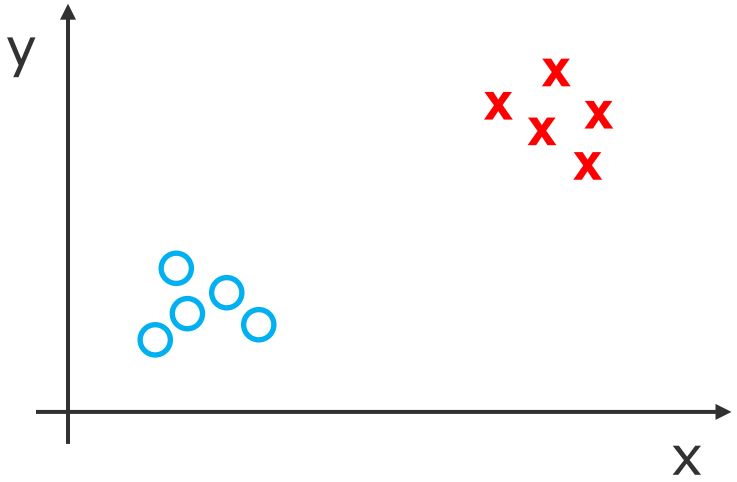


https://en.wikipedia.org/wiki/Apple
https://cdn4.vectorstock.com/i/1000x1000/16/58/robot-arm-line-icon-sign-on-vector-17841658.jpg

Unsupervised Learning

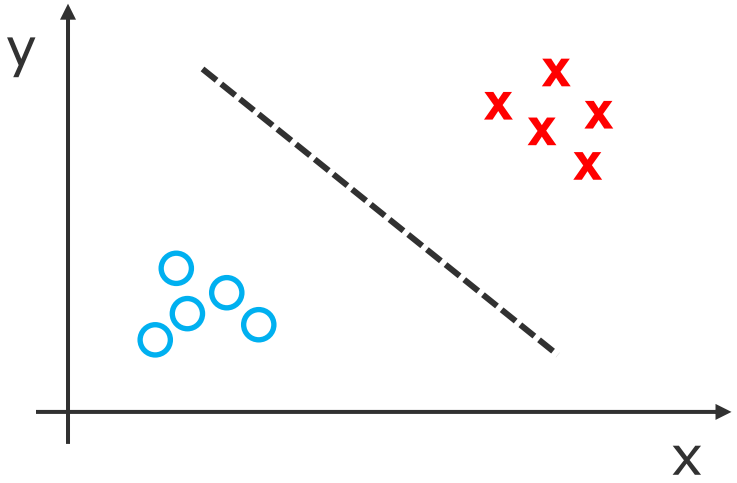
Data Management for Digital Health, Winter 2023

Supervised Learning



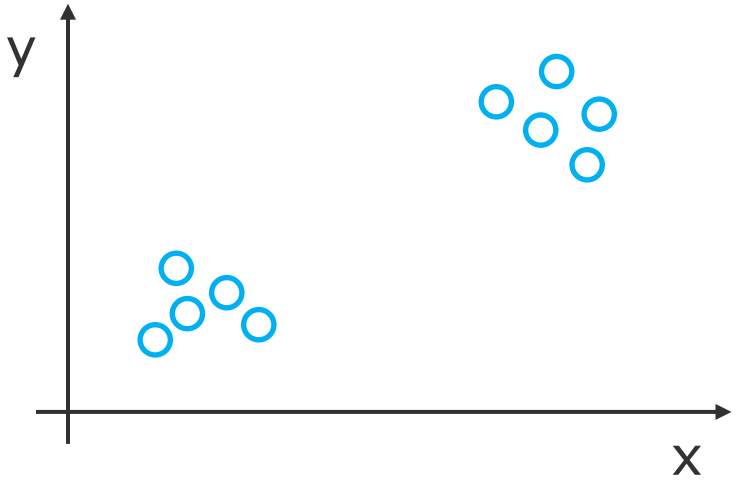
Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots, (x^{(n)}, y^{(n)})\}$

Supervised Learning



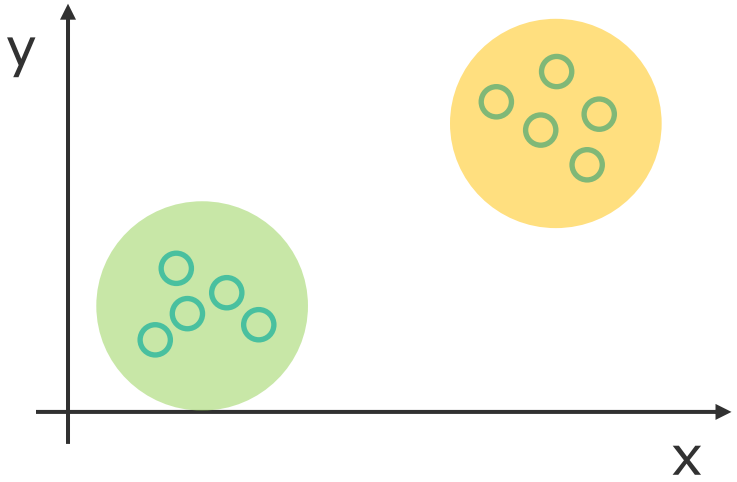
Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), \ldots, (x^{(n)}, y^{(n)})\}$

Unsupervised Learning



Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(n)}\}$

Unsupervised Learning

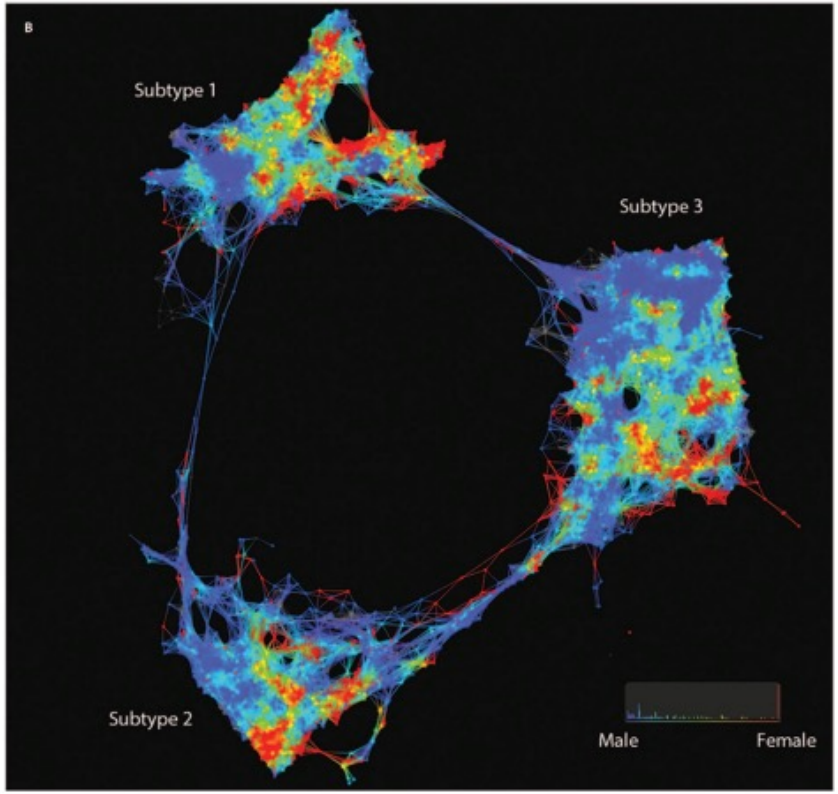


Clustering algorithm

Training set: $\{x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(n)}\}$

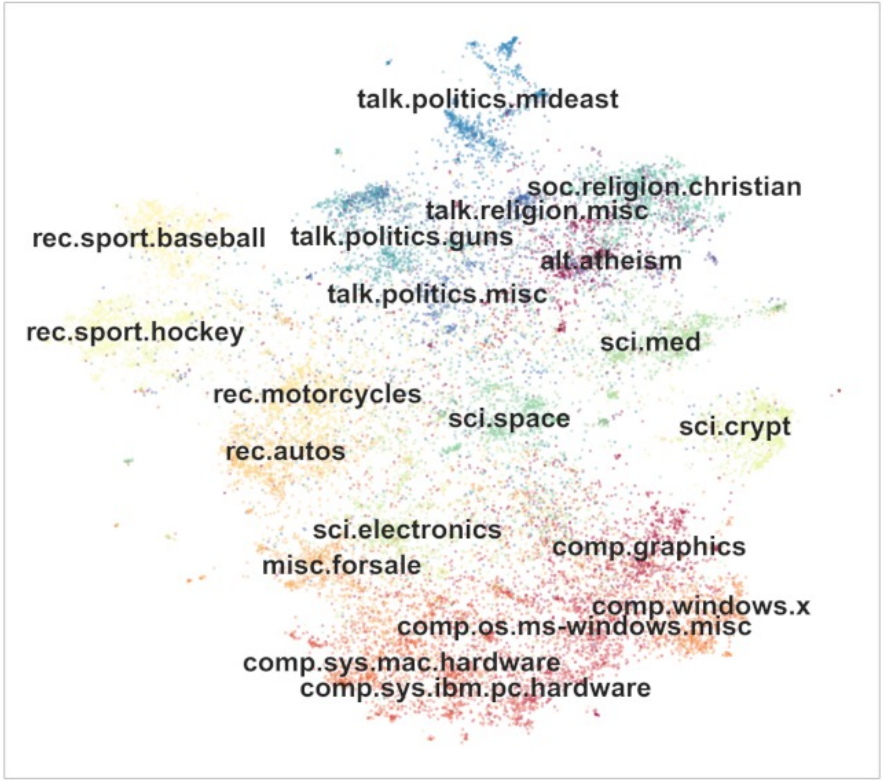
Clusters of Patients

(**B**) Patient-patient network for topology patterns on 2551 T2D patients. Each node represents a single or a group of patients with the significant similarity based on their clinical features. Edge connected with nodes indicates the nodes have shared patients. Red color represents the enrichment for patients with females, and blue color represents the enrichment for males.



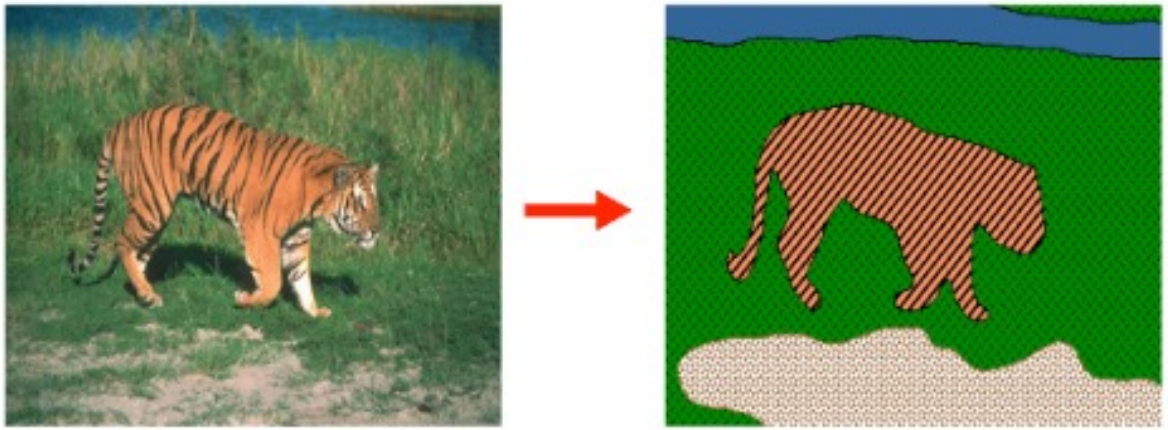
Li, Li, et al. "Identification of type 2 diabetes subgroups through topological analysis of patient similarity." *Science translational medicine* 7.311 (2015): 311ra174-311ra174.

Clusters of Text Documents

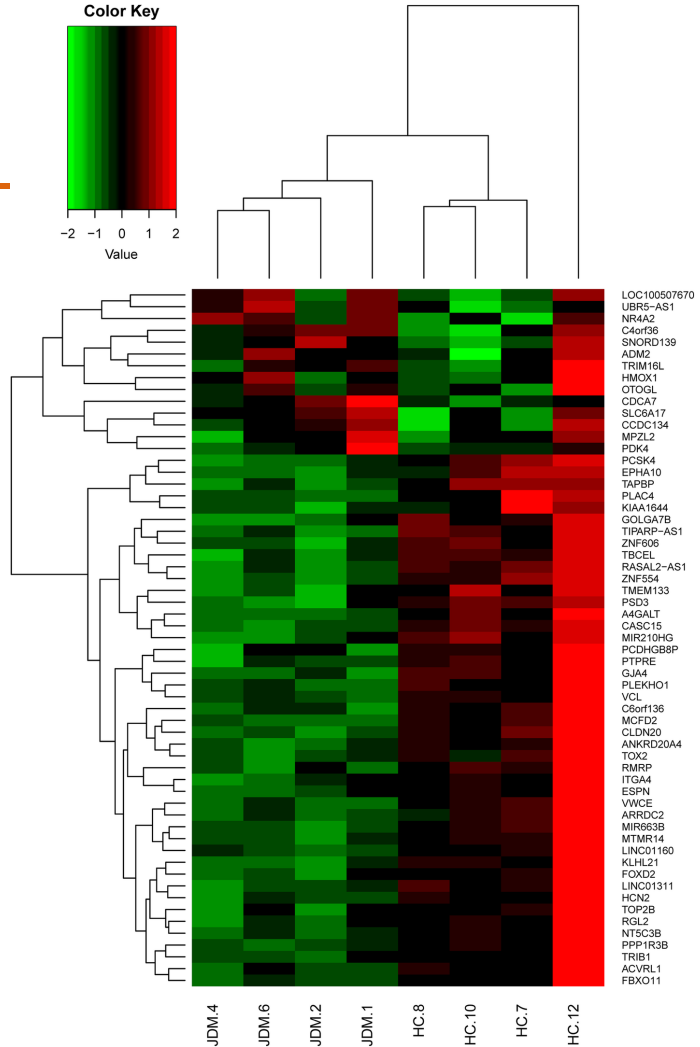


https://www.sbert.net/examples/applications/clustering/README.html

Clusters of Pixels



Clusters of Genes



Unsupervised hierarchical clustering analysis of gene expression for the 59 differentially expressed genes between JDM and HC. The heatmap shows the median-normalized expression of individual genes across all samples. Heatmap colors represent relative mRNA expression as indicated in the color key

Jiang, K., Karasawa, R., Hu, Z. *et al.* Plasma exosomes from children with juvenile dermatomyositis are taken up by human aortic endothelial cells and are associated with altered gene expression in those cells. *Pediatr Rheumatol* **17**, 41 (2019). https://doi.org/10.1186/s12969-019-0347-0

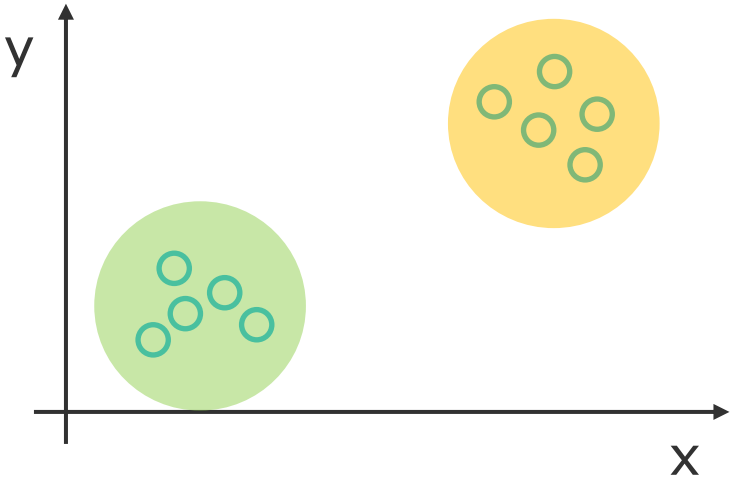
Hasso Plattner Institut

Unsupervised Learning

Data Management for Digital Health, Winter 2023

Clustering

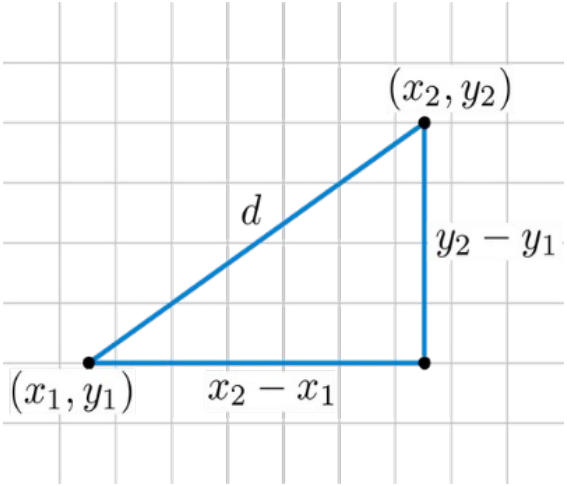
- Identifies sub-groups without explicit labels
- „Good“ clustering:
  - Similar data belong to the same cluster
  - Dissimilar data belong to different clusters
- How do we measure similarity?



Similarity and Distance Measures

- Distance measures D(X,Y) quantify similarity or dissimilarity between two data points X,Y
- Mathematical function determining how 'far apart' two entities are in the feature space
- Key properties:
  - Non-negativity $D(X,Y) > 0 \; if \; X \neq Y$
  - Identity of Indiscernibles $D(X,Y) = 0 \;\; iff \; X = Y$
  - Symmetry $D(X,Y) = D(Y,X)$
  - Triangle Inequality $D(X,Z) \leq D(X,Y) + D(Y,Z)$
- Similarity: 1 – D(X,Y)

Euclidean Distance

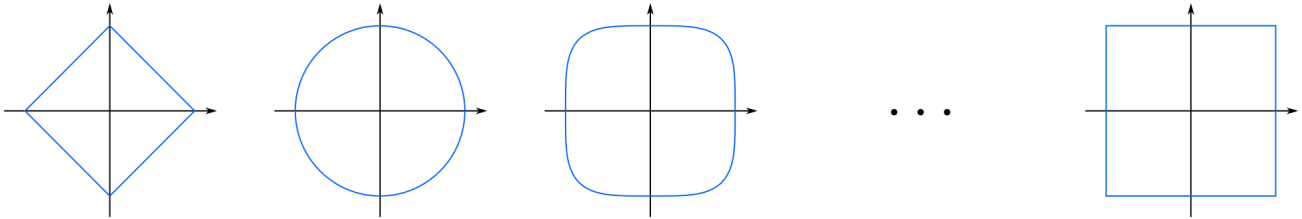


$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Minkowski Distance

- Generalized formula for distance in *n* dimensions
- P=2 Euclidian distance
- P=1 Manhattan distance
- P= ∞ Chebyshev distance

- $D(X,Y) = (\sum_{i=1}^{n}|x_i - y_i|^p)^{\frac{1}{p}}$
- $D(X,Y) = (\sum_{i=1}^{n}|x_i - y_i|^2)^{\frac{1}{2}}$
- $D(X,Y) = (\sum_{i=1}^{n}|x_i - y_i|^1)^1$
- $D(X,Y) = \lim_{n\to\infty}(\sum_{i=1}^{n}|x_i - y_i|^p)^{\frac{1}{p}} = \max(|x_i - y_i|)$



$p = 2^0 = 1$ $p = 2^1 = 2$ $p = 2^2 = 4$ $p = 2^\infty = \infty$

Minkowski Distance Example

- Suppose following two vectors represent three attributes of two data points:
  - X = (1, 0, 2)
  - Y= (0, 1, 0)

- Euclidian distance (P = 2)

$$\sqrt{(1-0)^2 + (0-1)^2 + (2-0)^2} = \sqrt{1+1+4} = \sqrt{6}$$

- Manhattan distance (P = 1)

$$|1-0| + |0-1| + |2-0| = |1| + |-1| + |2| = 4$$

- Chebyshev distance (P= ∞)

$$\max(1,1,2) = 2$$

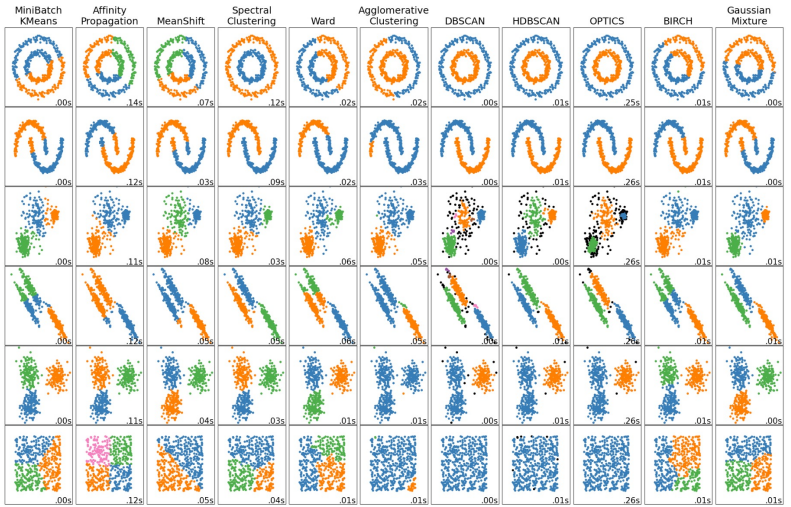
Other Similarity / Distance Measures

- Jaccard Coefficient $\frac{|X \cap Y|}{|X \cup Y|}$
- Cosine Similarity: $\frac{X \cdot Y}{\|X\| \|Y\|}$
- For strings:
  - Hamming distance (same length strings)
  - Edit distance

		s	i	t	t	i	n	g
	0	1	2	3	4	5	6	7
k	1	1	2	3	4	5	6	7
i	2	2	1	2	3	4	5	6
t	3	3	2	1	2	3	4	5
t	4	4	3	2	1	2	3	4
e	5	5	4	3	2	2	3	4
n	6	6	5	4	3	3	2	3

Clustering Algorithms

- Partitioning:
  - *k*-Means
  - Expectation-Maximization (EM) using Gaussian Mixture Models (GMM)
  - Density-Based Spatial Clustering of Applications with Noise (DBSCAN)
- Hierarchical:
  - Agglomerative Hierarchical Clustering



A comparison of the clustering algorithms in scikit-learn

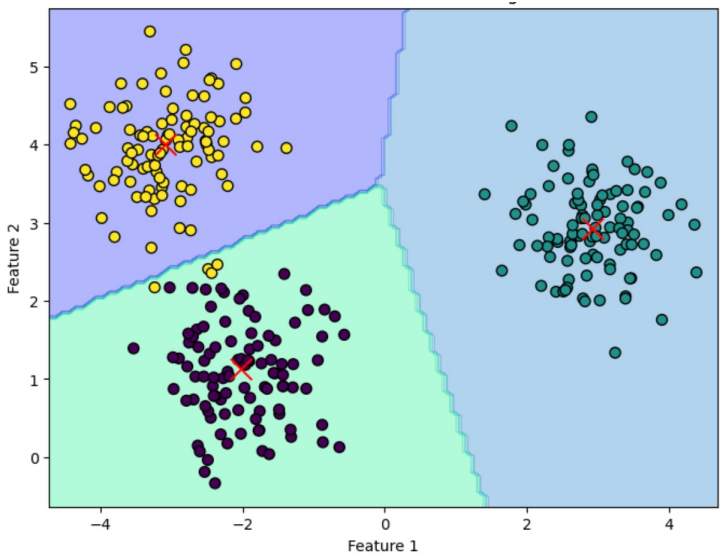
Unsupervised Learning

Data Management for Digital Health, Winter 2023

https://scikit-learn.org/stable/modules/clustering.html

k-Means Clustering Idea

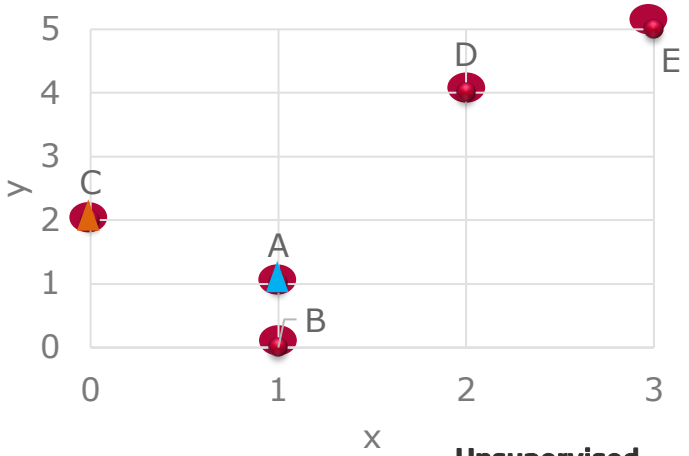
1. Initialize: Choose *k* examples (data points) from the dataset as initial centroids (randomly)
2. Cluster assignment: Data points that are the closest (similar) to a centroid will create a cluster
3. Move the centroid: A centroid's new value is going to be the mean of all the examples in a cluster
4. Repeating: Keep repeating step 2 and 3 until the centroids stop moving, in other words, *k*-Means algorithm is converged



k-Means Clustering
Step 0

- k = 2, Euclidean distance
- A and C are randomly selected as the initial means

		x	y
c_1	A	1	1
	B	1	0
c_2	C	0	2
	D	2	4
	E	3	5

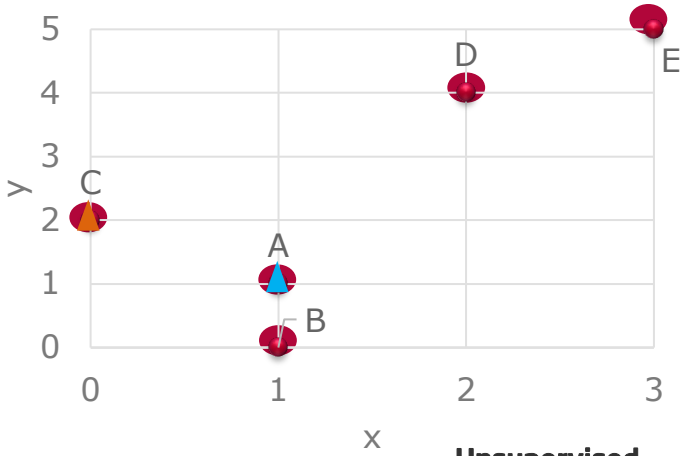


k-Means Clustering
Step 1.1

- Compute distances between each of the cluster means and all other points
- Assign nearest centroid to each point

		x	y
c_1	A	1	1
	B	1	0
c_2	C	0	2
	D	2	4
	E	3	5

	Distance to cluster	
	1	2
A	0	1.4
B	1	2.2
C	1.4	0
D	3.2	2.8
E	4.5	4.2



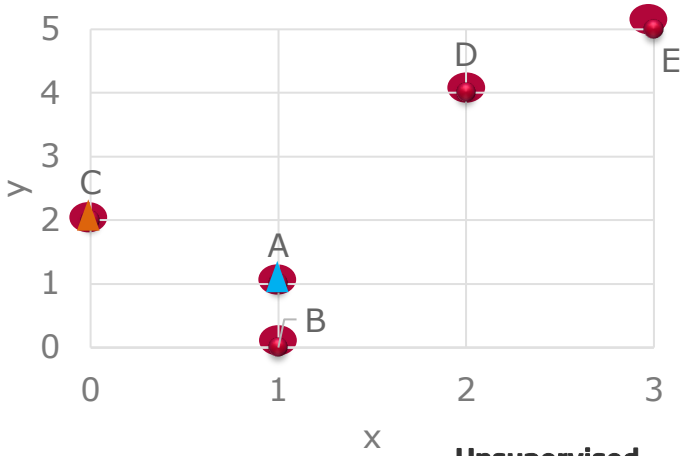
k-Means Clustering
Step 1.1

- Assign each case to the cluster having the closest mean
- Recalculate the cluster means

		x	y
c_1	A	1	1
	B	1	0
	C	0	2
c_2	D	2	4
	E	3	5

Distance to cluster

	1	2	Cluster
A	0	1.4	1
B	1	2.2	1
C	1.4	0	2
D	3.2	2.8	2
E	4.5	4.2	2



k-Means Clustering
Step 1.1

- Assign each case to the cluster having the closest mean
- Recalculate the cluster means

	x	y
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

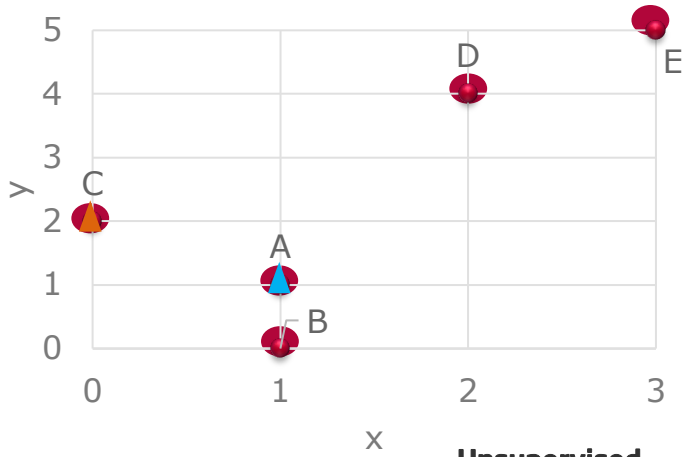
Distance to cluster

	1	2	Cluster
A	0	1.4	1
B	1	2.2	1
C	1.4	0	2
D	3.2	2.8	2
E	4.5	4.2	2

means

$c_1 = (1.0, 0.5)$

$c_2 = (1.7, 3.7)$



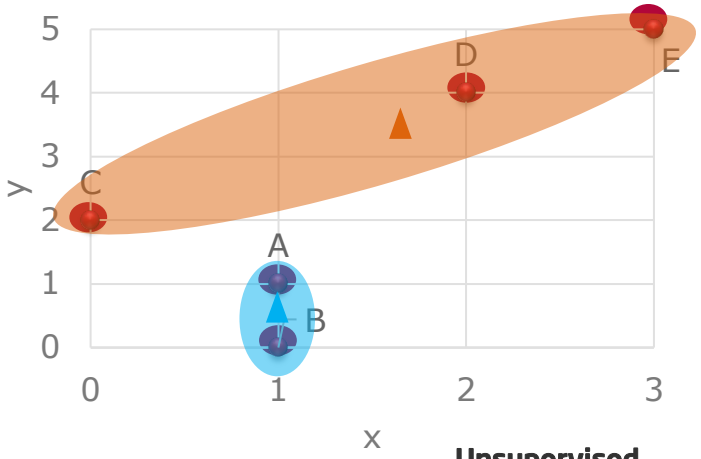
k-Means Clustering
Step 1.1 Plot

- Assign each case to the cluster having the closest mean
- Recalculate the cluster means

		x	y
c_1	A	1	1
	B	1	0
	C	0	2
c_2	D	2	4
	E	3	5

Distance to cluster

	1	2	Cluster
A	0	1.4	1
B	1	2.2	1
C	1.4	0	2
D	3.2	2.8	2
E	4.5	4.2	2



c_1 $c_1 = (1.0, 0.5)$

c_2 $c_2 = (1.7, 3.7)$

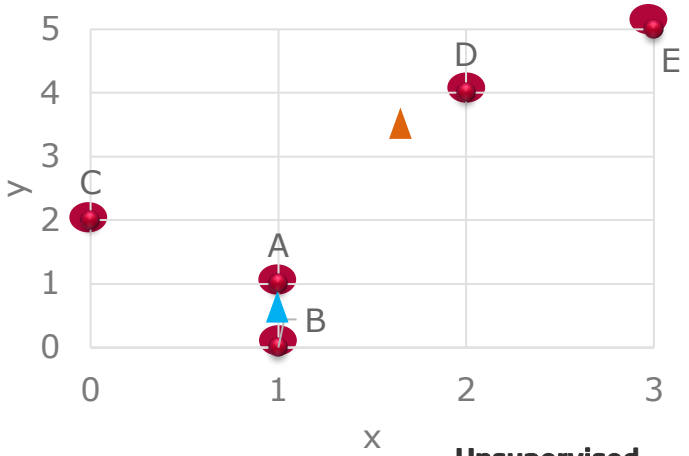
k-Means Clustering
Step 2.1

- Compute distances between each of the cluster and all other points

	x	y
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

Distance to cluster

	1	2
A	0.5	2.7
B	0.5	3.7
C	1.8	2.4
D	3.6	0.5
E	4.9	1.9



c_1 $c_1 = (1.0, 0.5)$

c_2 $c_2 = (1.7, 3.7)$

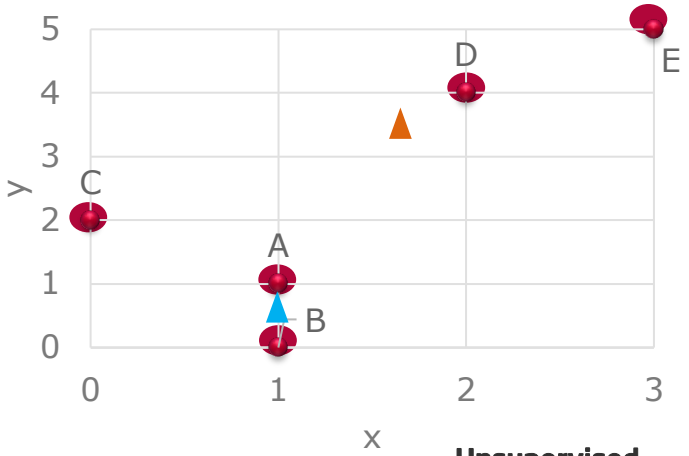
k-Means Clustering
Step 2.1

- Compute distances between each of the cluster and all other points

	x	y
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

Distance to cluster

	1	2
A	0.5	2.7
B	0.5	3.7
C	1.8	2.4
D	3.6	0.5
E	4.9	1.9



c_1 = (1.0, 0.5)

c_2 = (1.7, 3.7)

k-Means Clustering
Step 2.1

- Compute distances between each of the cluster and all other points

	x	y
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

c_1

c_2

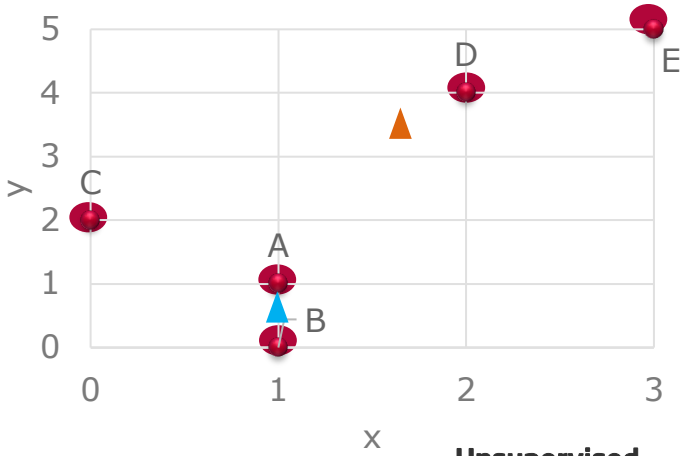
Distance to cluster

	1	2	Cluster
A	0.5	2.7	1
B	0.5	3.7	1
C	1.8	2.4	1
D	3.6	0.5	2
E	4.9	1.9	2

means

c_1 $c_1 = (0.7, 1.0)$

c_2 $c_2 = (2.5, 4.5)$



Unsupervised Learning

k-Means Clustering
Step 2.1 Plot

- Assign each case to the cluster having the closest mean
- Recalculate the cluster means

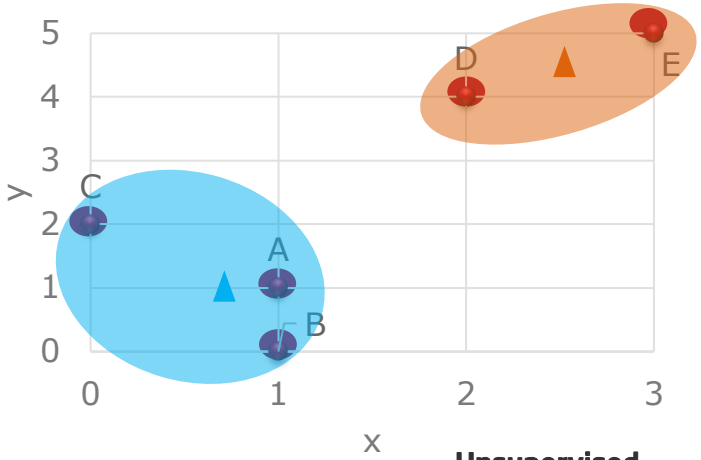
	x	y
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

Distance to cluster

	1	2	Cluster
A	0.5	2.7	1
B	0.5	3.7	1
C	1.8	2.4	1
D	3.6	0.5	2
E	4.9	1.9	2

c_1 $c_1 = (0.7, 1.0)$

c_2 $c_2 = (2.5, 4.5)$



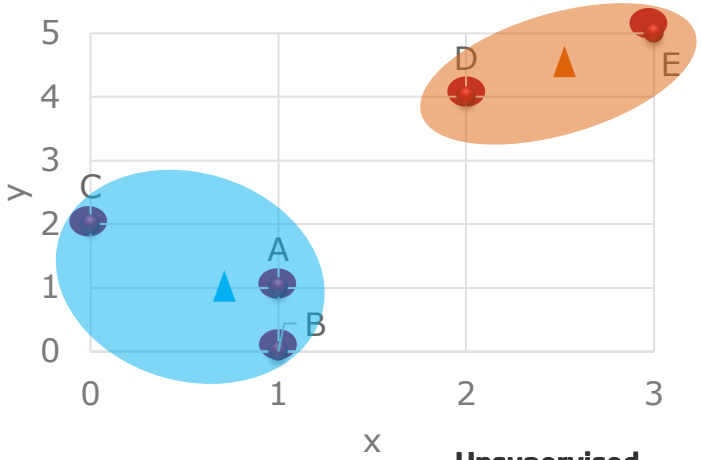
k-Means Clustering
Step 3

- Algorithm has converged – re-calculating distances, reassigning cases to clusters results in no change
- This is the final solution

	x	y
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

Distance to cluster

	1	2	Cluster
A	0.5	2.7	1
B	0.5	3.7	1
C	1.8	2.4	1
D	3.6	0.5	2
E	4.9	1.9	2



c_1 = (0.7, 1.0)

c_2 = (2.5, 4.5)

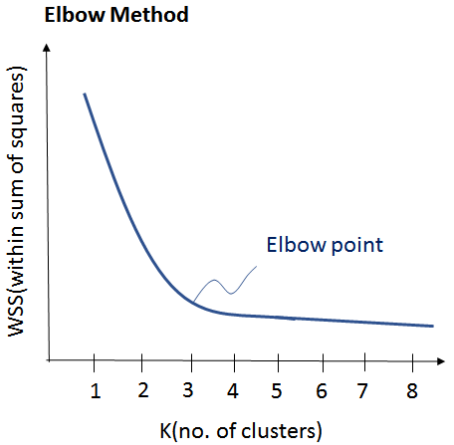
k-Means Clustering Algorithm

- Inputs: K, set of points $x_1, ..., x_n$
- Place centroids $c_1, ..., c_K$ at random locations
- Repeat until convergence:
  - For each point x_i:
    - Find nearest centroid c_j $\quad arg\min_j D(x_i, c_j)$
    - Assign the point x_i to cluster j
  - For each cluster j = 1, ..., K: $\quad c_j(a) = \frac{1}{n_j}\sum_{x_i \to c_j} x_i(a) \quad for\ a = 1..d$
    - New centroid c_j = mean of all points x_i assigned to cluster j in previous step
- Stop when none of the cluster assignments change
- Variants: k-medians, k-medoids

Distance (e.g. Euclidian between instance x_i and cluster c_j)

What should *k* be?

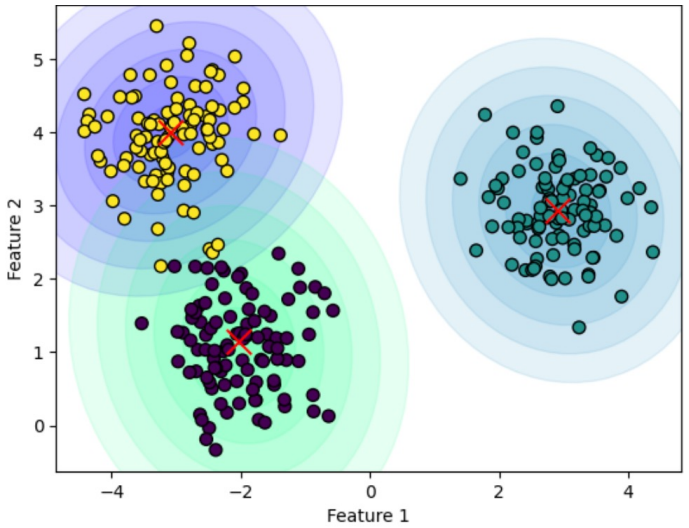
- *k*-means clustering algorithm requires the number of clusters *k* set
- What is the magic number *k?*
- One heuristic is called Elbow Curve
  - Train *k*-Means models for different numbers of *k*
  - y axis := sum of the square distance between points in a cluster and its centroid
  - Stop when returns are diminishing (overfitting)
  - … not very accurate and often subjective



https://www.edureka.co/blog/k-means-clustering-algorithm/

Gaussian Mixture Models (GMMs)

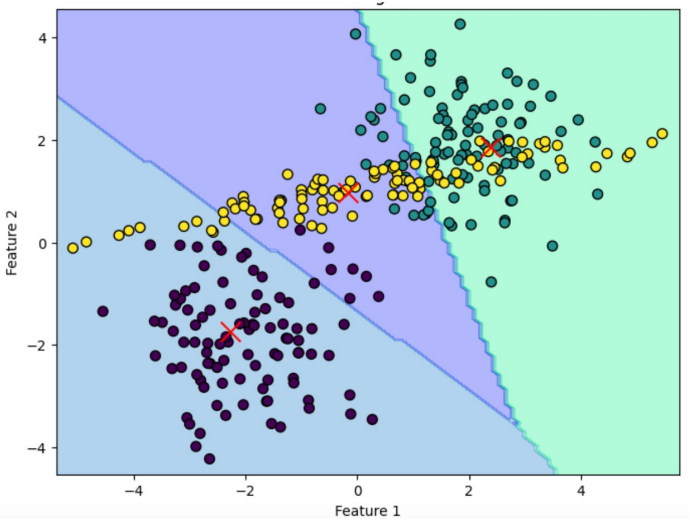
- „Soft“ variant of k-Means
- Uses a mixture of k Gaussian distributions
- Each cluster is defined by mean and covariance
- Instead of fixed cluster assignment, each data point has some likelihood of belonging to each cluster
- Iterative estimation of Gaussian parameters similar to *k*-Means

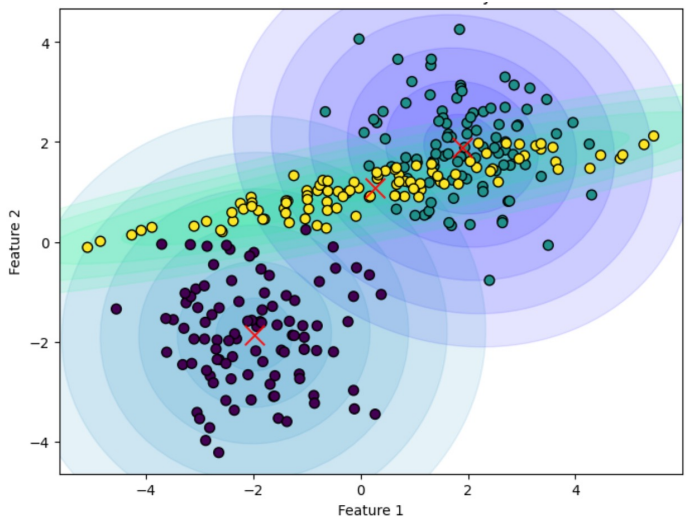


Expectation-Maximization (EM) for GMMs

1. **Initialization:** Select the number of clusters and randomly initialize the Gaussian distribution parameters (mean, variance) for each one of them
2. **E-step**: Calculate probability of each data point belonging to a particular cluster (The closer the point is to the Gaussian's center, the better are the chances of it belonging to the cluster)
3. **M-step:** Update parameters of the Gaussian distributions (means, covariances, and mixture weights) to maximize the likelihood of the observed data
4. **Convergence**: Repeat the steps 2 and 3 until convergence

Gaussian Mixture Models Considerations

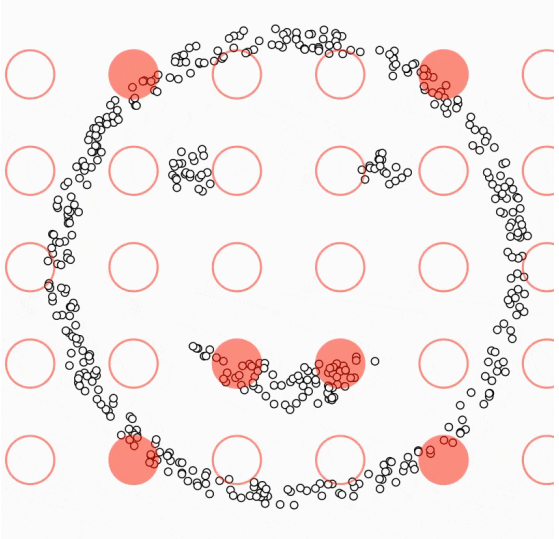




- GMMs supports cluster shapes that are not spherical
- Number of clusters still needs to be chosen a priori
- Training is rather slow, but means can be initialized from *k*-Means

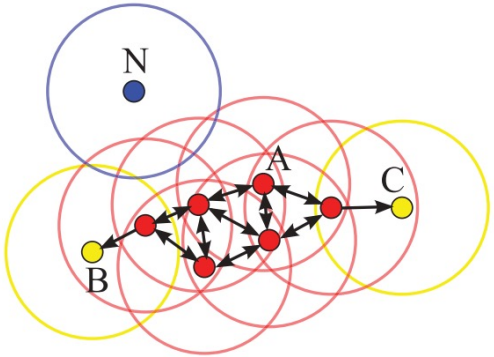
Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

- Concepts
  - Core Points
  - Border Points
  - Noise Points
- Two parameters
  - minPts := Minimum number of point needed in a cluster
  - epsilon := Radius to assign a point to cluster using distance function



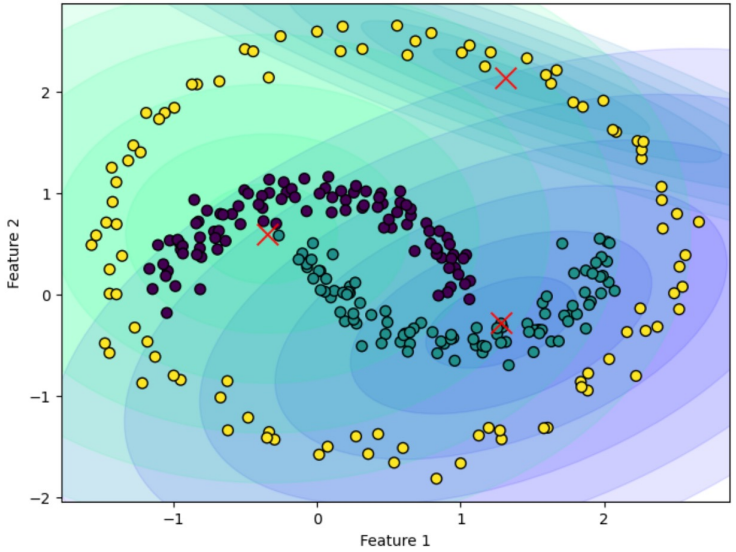
DBSCAN Pseudocode

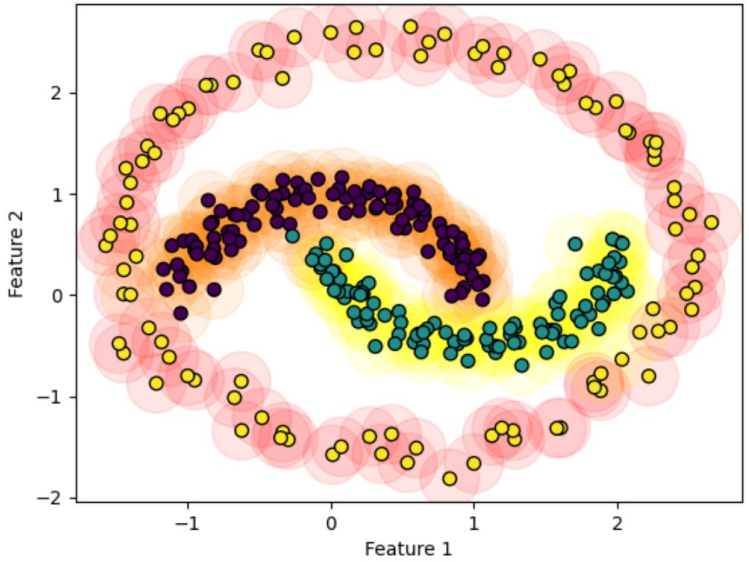
```
for each unvisited point P in the dataset:
    mark P as visited
    N := neighbors of P within epsilon distance.
    if size of N < minPts:
        mark P as noise
    else:
        create a new cluster with P as a core point
        for each neighbor P' in N:
            if P' previously marked as noise:
                include P' in cluster as border point
            if P' has been visited:
                continue
            include P' in current cluster as core point
            N' := neighbors of P' within epsilon distance
            if size of N' > minPts:
                expand N by N'
```



Schubert, Erich, et al. "DBSCAN revisited, revisited: why and how you should (still) use DBSCAN." *ACM Transactions on Database Systems (TODS)* 42.3 (2017): 1-21.

DBSCAN Considerations

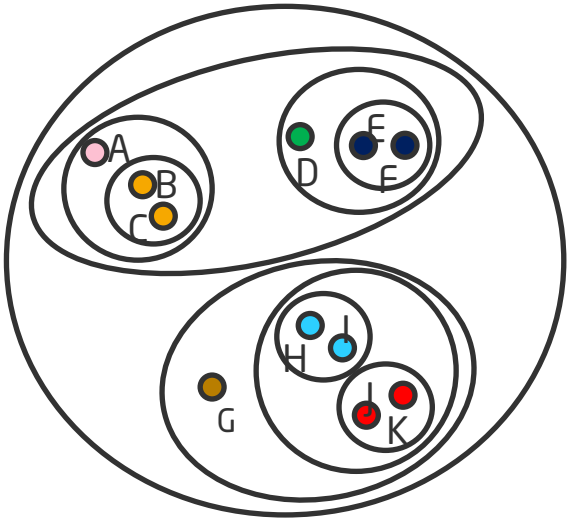


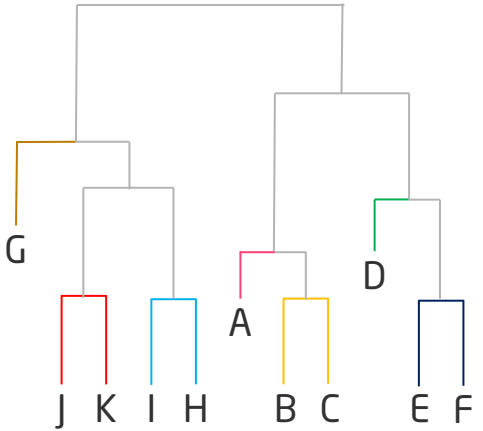


- Explicit handling of noise
- Arbitrary cluster shapes
- Quite popular, efficient implementations available
- Sensitive to choice of parameters, especially epsilon

Hierarchical Clustering

- Builds a hierarchy of clusters
- Agglomerative (bottom-up)
- Divisive (top-down)
- Results can be presented as a dendrogram

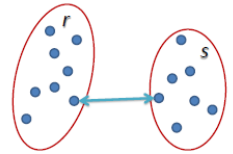




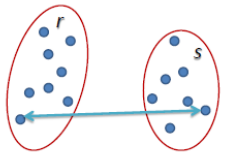
https://www.statisticshowto.com/hierarchical-clustering/

Agglomerative Hierarchical Clustering Overview

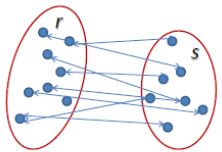
- Each object is a member of a hierarchy of clusters
- At the bottom of the hierarchy each object is a single cluster
- At the top of the hierarchy all objects belong to single clusters
- Clusters can be linked using different strategies, e.g.:
  - Single Linkage: Minimimizes distance between closest observation
  - Maximum or complete linkage: Minimizes the maximum distance between observations
  - Average linkage: Minimizes the average of the distance between all observation



$$L(r,s) = \min(D(x_{ri}, x_{sj}))$$



$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$

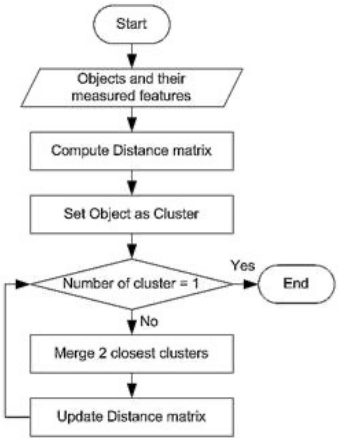


$$L(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$

https://www.saedsayad.com/clustering_hierarchical.htm

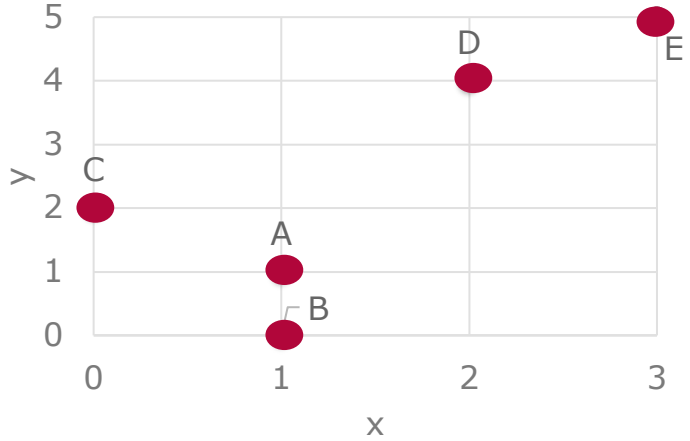
Agglomerative Hierarchical Clustering Idea

1. Convert object features to distance matrix
2. Set each object as a cluster (thus if there are 5 objects, there will be 5 clusters in the beginning)
3. Iterate until number of clusters is 1
   a. Merge two closest clusters
   b. Update distance matrix



https://people.revoledu.com/kardi/tutorial/Clustering/Hierarchical%20Clustering%20Algorithm.htm

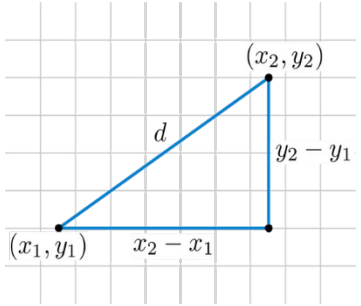
Agglomerative Hierarchical Clustering
Fill Distance Matrix



	x	y
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

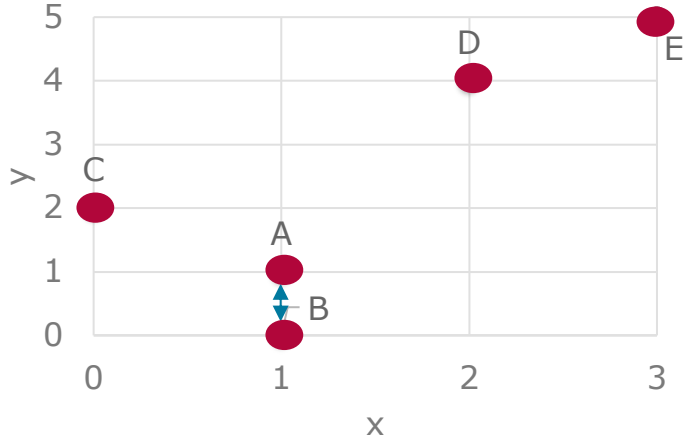
Euclidean distances

	A	B	C	D	E
A	0				
B		0			
C			0		
D				0	
E					0



$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

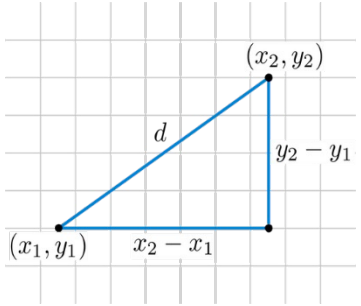
Agglomerative Hierarchical Clustering
Fill Distance Matrix



	x	y
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

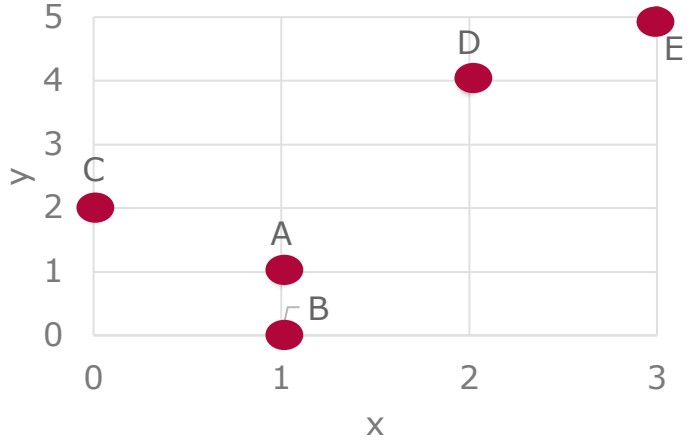
Euclidean distances

	A	B	C	D	E
A	0				
B	?	0			
C			0		
D				0	
E					0



$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

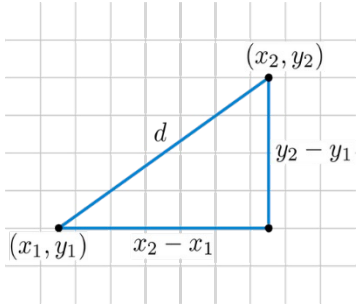
Agglomerative Hierarchical Clustering
Fill Distance Matrix



	x	y
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

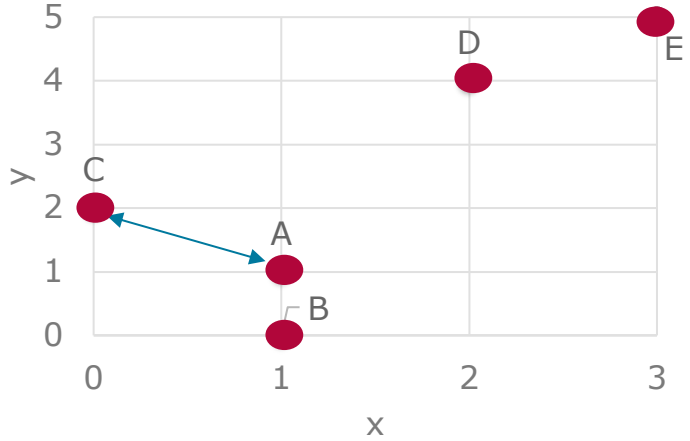
Euclidean distances

	A	B	C	D	E
A	0				
B	1	0			
C			0		
D				0	
E					0



$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

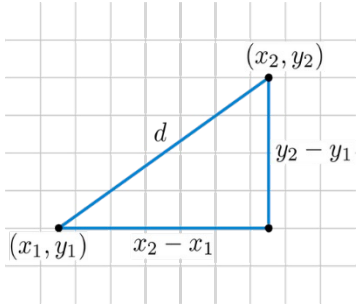
Agglomerative Hierarchical Clustering
Fill Distance Matrix



	x	y
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

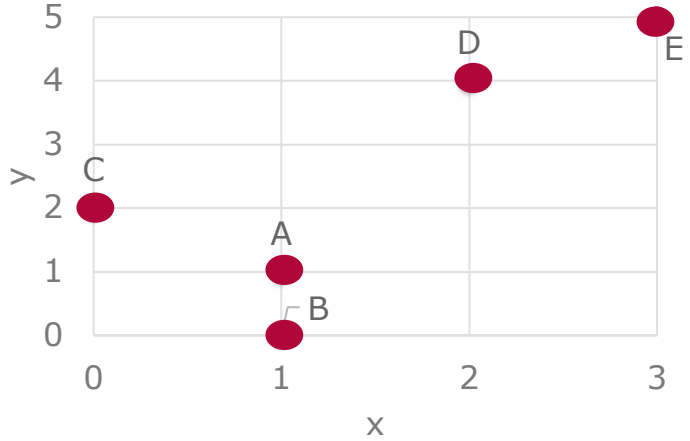
Euclidean distances

	A	B	C	D	E
A	0				
B	1	0			
C	?		0		
D				0	
E					0



$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

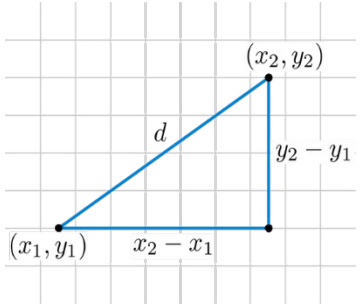
Agglomerative Hierarchical Clustering
Fill Distance Matrix



	x	y
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

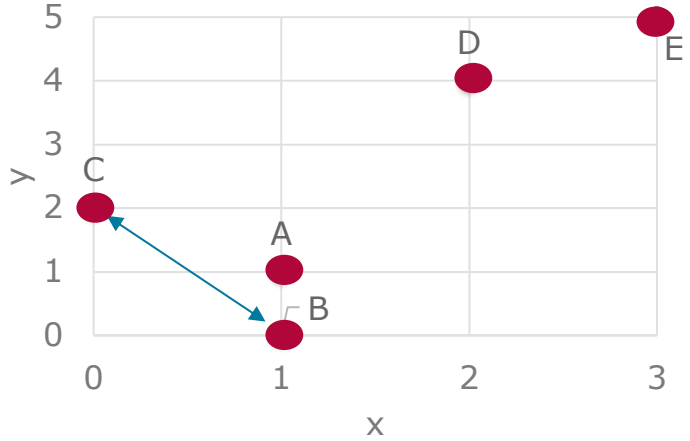
Euclidean distances

	A	B	C	D	E
A	0				
B	1	0			
C	1.4		0		
D				0	
E					0



$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

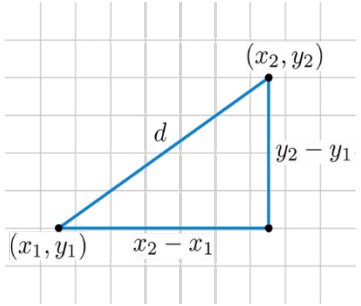
Agglomerative Hierarchical Clustering
Fill Distance Matrix



	x	y
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

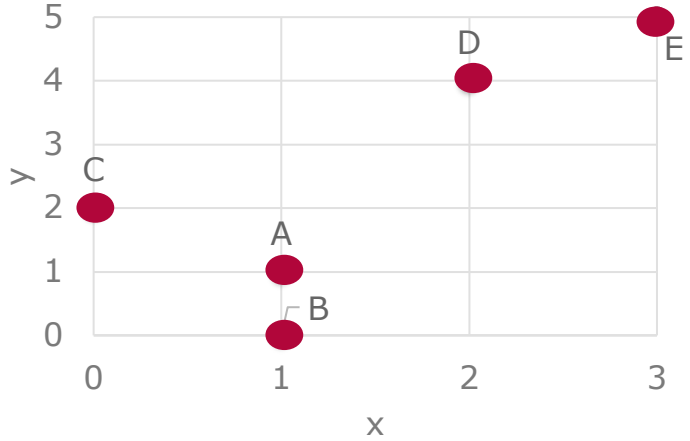
Euclidean distances

	A	B	C	D	E
A	0				
B	1	0			
C	1.4	?	0		
D				0	
E					0



$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

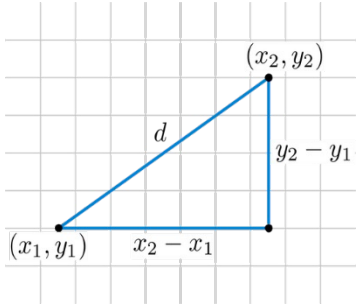
Agglomerative Hierarchical Clustering
Fill Distance Matrix



	x	y
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

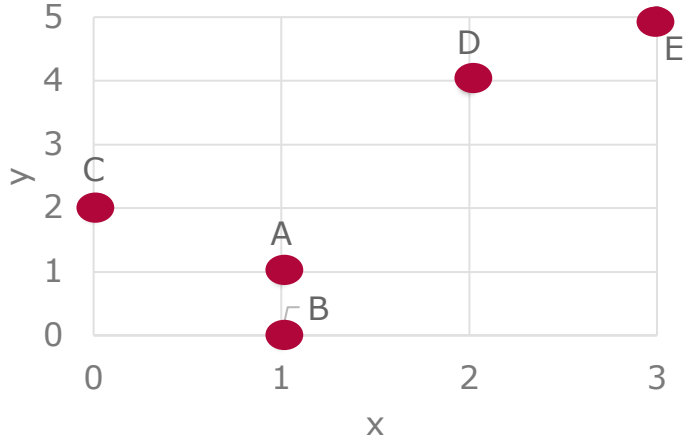
Euclidean distances

	A	B	C	D	E
A	0				
B	1	0			
C	1.4	2.2	0		
D				0	
E					0



$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

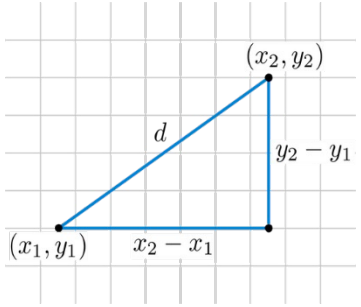
Agglomerative Hierarchical Clustering
Fill Distance Matrix



	x	y
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

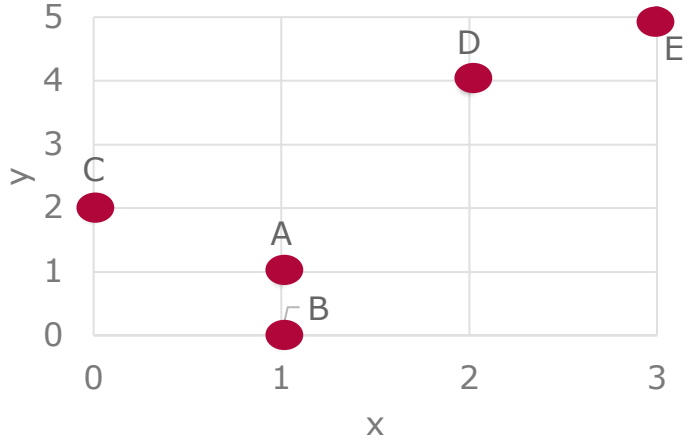
Euclidean distances

	A	B	C	D	E
A	0				
B	1	0			
C	1.4	2.2	0		
D	3.2	4.1	2.8	0	
E	4.5	5.4	4.2	1.4	0



$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

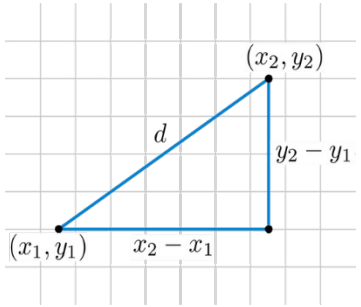
Agglomerative Hierarchical Clustering
Fill Distance Matrix



	x	y
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

Euclidean distances

	A	B	C	D	E
A	0	1	1.4	3.2	4.5
B	1	0	2.2	4.1	5.4
C	1.4	2.2	0	2.8	4.2
D	3.2	4.1	2.8	0	1.4
E	4.5	5.4	4.2	1.4	0

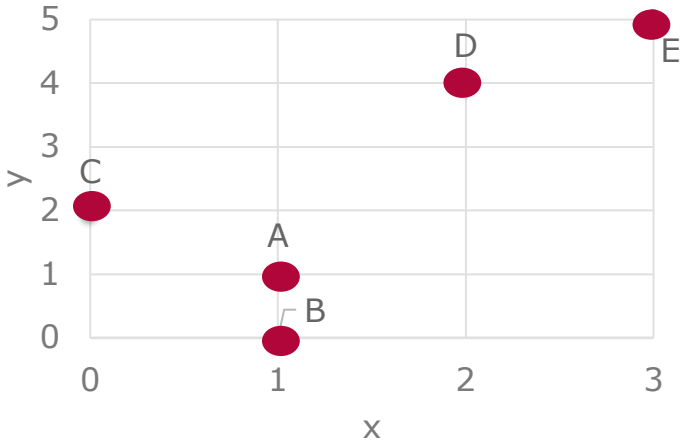


$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Agglomerative Hierarchical Clustering
Merge Clusters

- Join the two closest points into a cluster

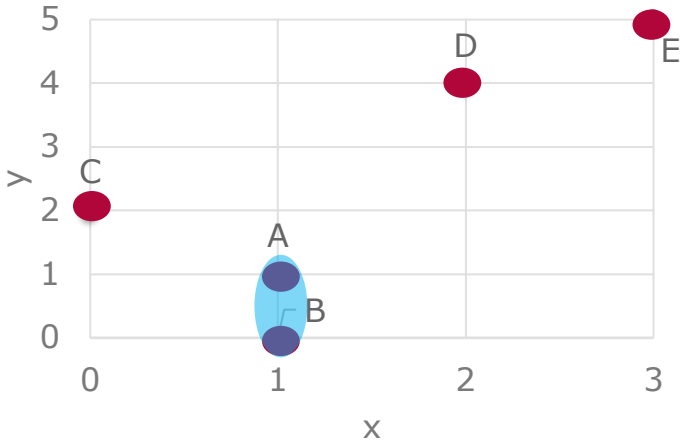
	A	B	C	D	E
A	0	1	1.4	3.2	4.5
B	1	0	2.2	4.1	5.4
C	1.4	2.2	0	2.8	4.2
D	3.2	4.1	2.8	0	1.4
E	4.5	5.4	4.2	1.4	0



Agglomerative Hierarchical Clustering
Merge Clusters

- Join the two closest points into a cluster

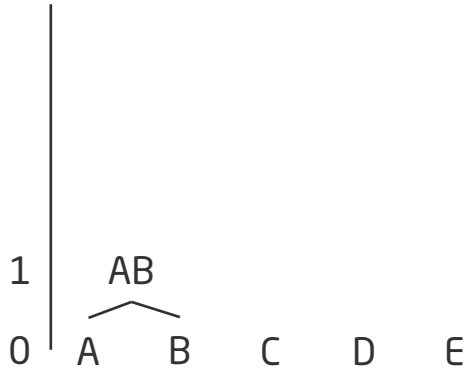
	A	B	C	D	E
A	0	1	1.4	3.2	4.5
B	1	0	2.2	4.1	5.4
C	1.4	2.2	0	2.8	4.2
D	3.2	4.1	2.8	0	1.4
E	4.5	5.4	4.2	1.4	0



Agglomerative Hierarchical Clustering
Update Distance Matrix

- Reduce the distance matrix, using the linkage methods (here: average)
- Draw the dendrogram

	AB	C	D	E
AB	0	1.8	3.6	4.9
C	1.8	0	2.8	4.2
D	3.6	2.8	0	1.4
E	4.9	4.2	1.4	0



	A	B	C	D	E
A	0	1	1.4	3.2	4.5
B	1	0	2.2	4.1	5.4
C	1.4	2.2	0	2.8	4.2
D	3.2	4.1	2.8	0	1.4
E	4.5	5.4	4.2	1.4	0

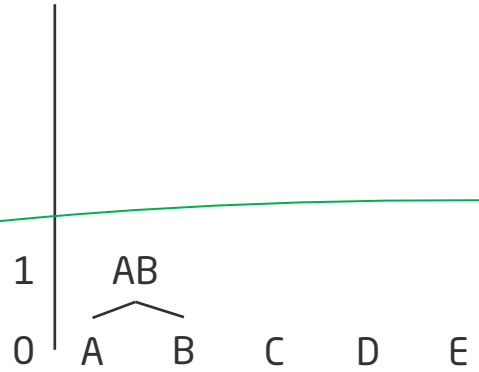
Agglomerative Hierarchical Clustering
Update Distance Matrix

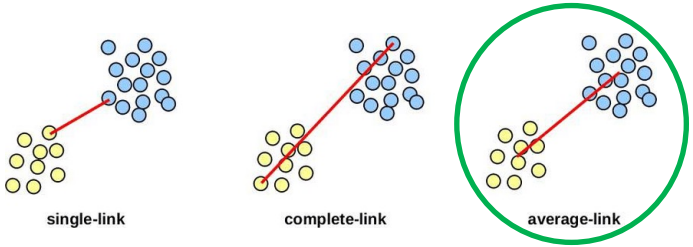
- Reduce the distance matrix, using the linkage methods (here: average)
- Draw the dendrogram

	A	B	C	D	E
A	0	1	1.4	3.2	4.5
B	1	0	2.2	4.1	5.4
C	1.4	2.2	0	2.8	4.2
D	3.2	4.1	2.8	0	1.4
E	4.5	5.4	4.2	1.4	0

average

	AB	C	D	E
AB	0	1.8	3.6	4.9
C	1.8	0	2.8	4.2
D	3.6	2.8	0	1.4
E	4.9	4.2	1.4	0

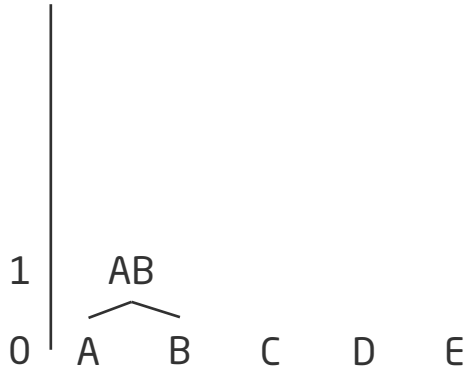




Agglomerative Hierarchical Clustering
Update Distance Matrix

- Reduce the distance matrix, using the linkage methods (here: average)
- Draw the dendrogram

	AB	C	D	E
AB	0	1.8	3.6	4.9
C	1.8	0	2.8	4.2
D	3.6	2.8	0	1.4
E	4.9	4.2	1.4	0

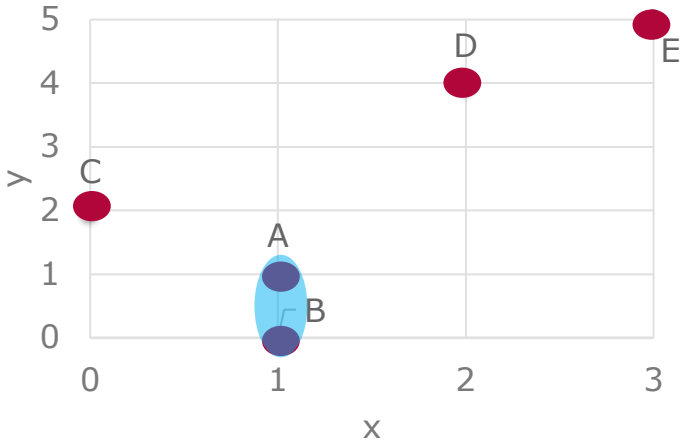


	A	B	C	D	E
A	0	1	1.4	3.2	4.5
B	1	0	2.2	4.1	5.4
C	1.4	2.2	0	2.8	4.2
D	3.2	4.1	2.8	0	1.4
E	4.5	5.4	4.2	1.4	0

Agglomerative Hierarchical Clustering
Merge Clusters

- Join the two closest points into a cluster

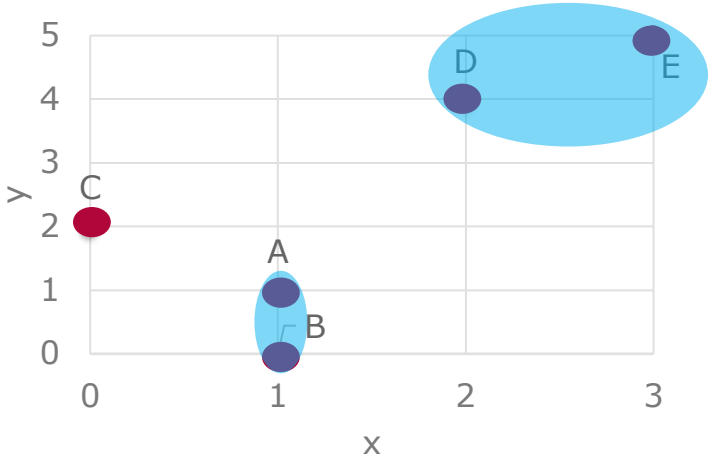
	AB	C	D	E
AB	0	1.8	3.6	4.9
C	1.8	0	2.8	4.2
D	3.6	2.8	0	1.4
E	4.9	4.2	1.4	0



Agglomerative Hierarchical Clustering
Merge Clusters

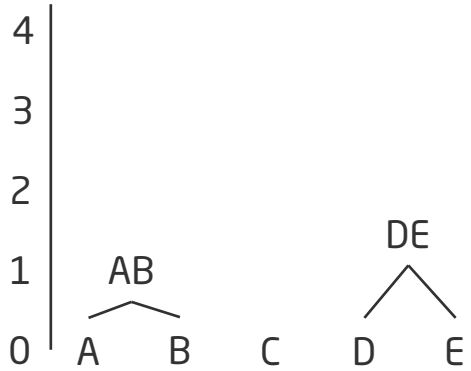
- Join the two closest points into a cluster

	AB	C	D	E
AB	0	1.8	3.6	4.9
C	1.8	0	2.8	4.2
D	3.6	2.8	0	1.4
E	4.9	4.2	1.4	0



Agglomerative Hierarchical Clustering
Update Distance Matrix

- Reduce the distance matrix, using the linkage methods
- Draw the dendrogram

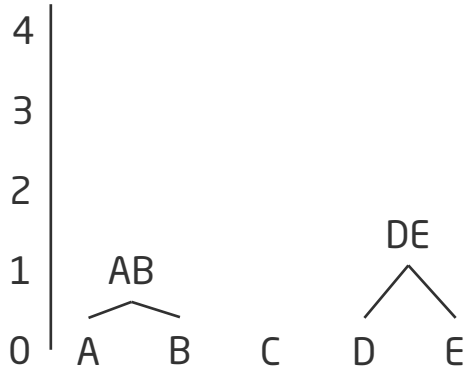


	AB	C	DE
AB	0	1.8	4.3
C	1.8	0	3.5
DE	4.3	3.5	0

	AB	C	D	E
AB	0	1.8	3.6	4.9
C	1.8	0	2.8	4.2
D	3.6	2.8	0	1.4
E	4.9	4.2	1.4	0

Agglomerative Hierarchical Clustering
Update Distance Matrix

- Reduce the distance matrix, using the linkage methods
- Draw the dendrogram



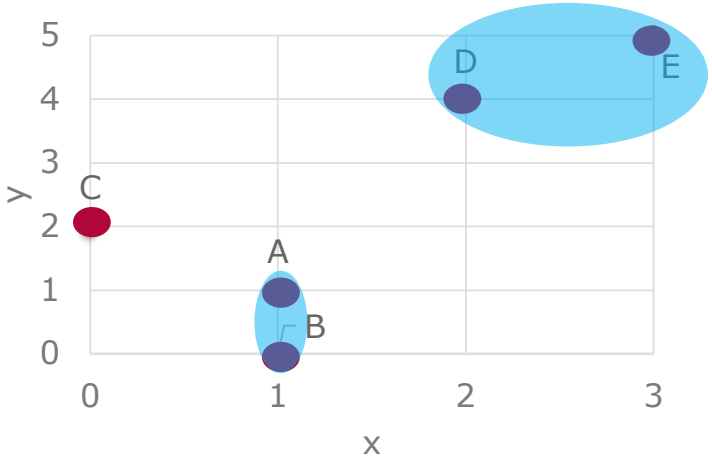
	AB	C	DE
AB	0	1.8	4.3
C	1.8	0	3.5
DE	4.3	3.5	0

	AB	C	D	E
AB	0	1.8	3.6	4.9
C	1.8	0	2.8	4.2
D	3.6	2.8	0	1.4
E	4.9	4.2	1.4	0

Agglomerative Hierarchical Clustering
Merge Clusters

- Join the two closest points into a cluster

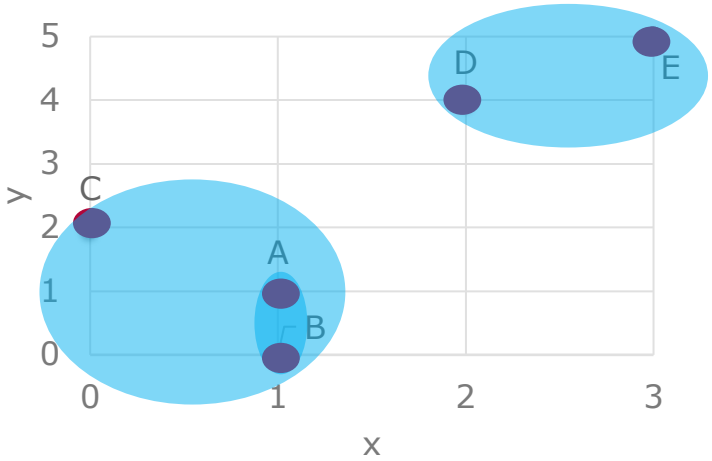
	AB	C	DE
AB	0	1.8	4.3
C	1.8	0	3.5
DE	4.3	3.5	0



Agglomerative Hierarchical Clustering
Merge Clusters

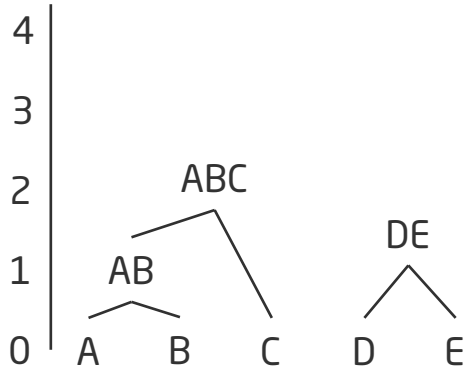
- Join the two closest points into a cluster

	AB	C	DE
AB	0	1.8	4.3
C	1.8	0	3.5
DE	4.3	3.5	0



Agglomerative Hierarchical Clustering
Update Distance Matrix

- Reduce the distance matrix, using the linkage methods
- Draw the dendrogram



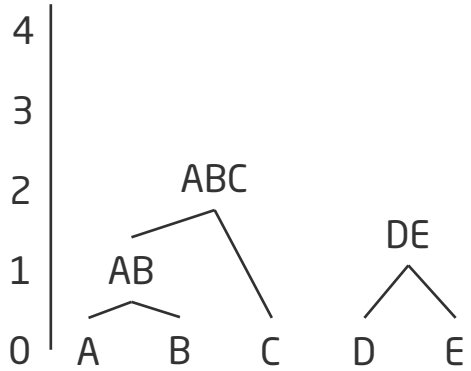
	AB	C	DE
AB	0	1.8	4.3
C	1.8	0	3.5
DE	4.3	3.5	0

	ABC	DE
ABC	0	3.9
DE	3.9	0

- Average linkage used

Agglomerative Hierarchical Clustering
Update Distance Matrix

- Reduce the distance matrix, using the linkage methods
- Draw the dendrogram



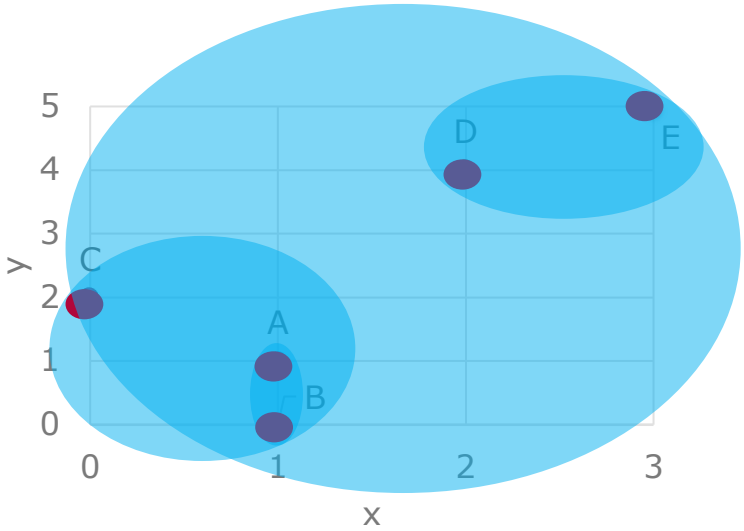
	AB	C	DE
AB	0	1.8	4.3
C	1.8	0	3.5
DE	4.3	3.5	0

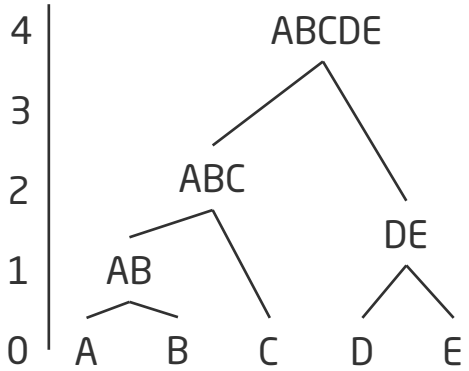
	ABC	DE
ABC	0	3.9
DE	3.9	0

- Average linkage used

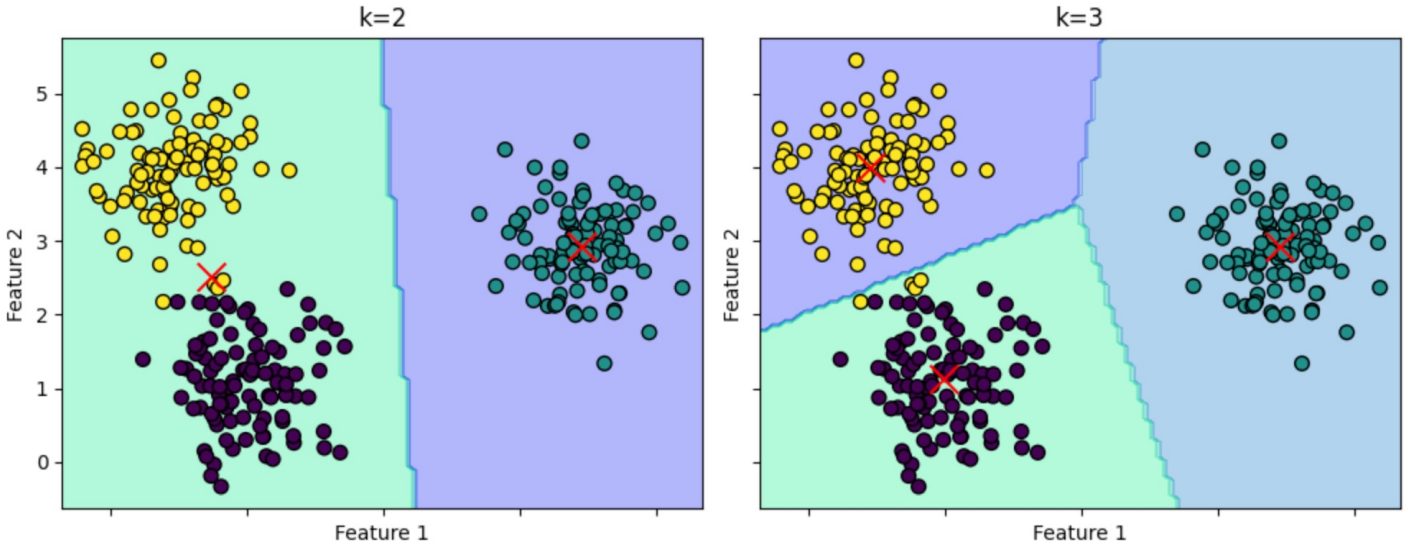
Agglomerative Hierarchical Clustering
Merge Clusters

- Join last two clusters





Evaluation of Clustering Results

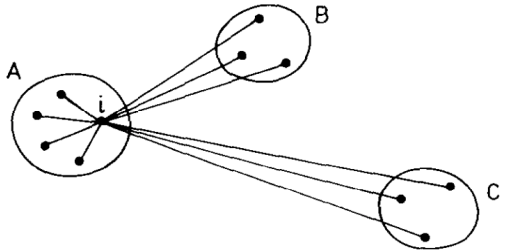


Intrinsic Evaluation

- Evaluation based on shape of clusters themselves
- e.g., Silhouette Coefficient:
  - for each object i in cluster A:
    - $s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$ (silhouette of object i), where
    - $a(i)$ = average dissimilarity of i to all other objects of A
    - $d(i, C)$ = average dissimilarity of i to all objects of C
    - $b(i) = \min_{A \neq C} d(i, C)$ (second best cluster for object i)
  - Silhouette coefficient for a particular clustering is the mean silhouette for all samples
- Other options: Davies-Bouldin index, Dunn Index, ...

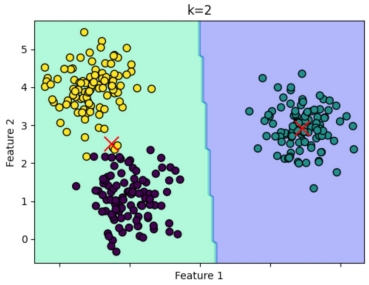
Journal of Computational and Applied Mathematics 20 (1987) 53–65
North-Holland

Silhouettes: a graphical aid to the interpretation and validation of cluster analysis

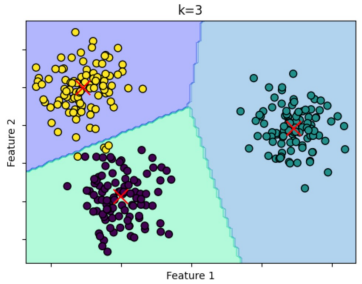


Extrinsic Evaluation

- Evaluation of the ability of clustering algorithms to separate class compared to ground truth
- Contingency Matrix
  - Similar to confusion matrix
  - How often do assignment to cluster and actual class occur together?



	Cluster 1	Cluster 2
Label 1	0	100
Label 2	100	0
Label 3	0	100



	Cluster 1	Cluster 2	Cluster 3
Label 1	100	0	0
Label 2	0	100	0
Label 3	4	0	96

Extrinsic Evaluation
Rand Index

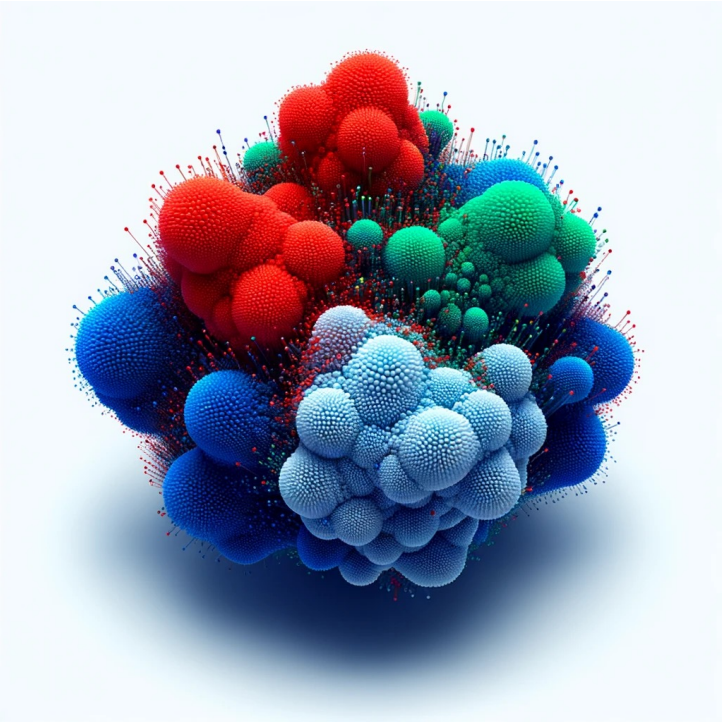
- Rand Index: Measures the agreement of all pairs of samples (similar to accuracy for classification)

$$R = \frac{TP + TN}{\binom{n}{2}}$$

  - TP is the number of pairs of points that are clustered together in the predicted and the ground truth partitioning
  - TN is the number of pairs of points that are assigned to different clusters in the predicted and the ground truth partitioning
  - $\binom{n}{2}$ is the number of pairs in a datasize of size n (TP + TN + FP + FN)

- Adjusted Rand index accounts for agreement by chance
- Others: mutual information, purity, ...

What to Take Home



- Clustering: art or science?
- Distance and similarity measures
- Clustering algorithms (*k*-Means, GMM, DBSCAN, Hierarchical)
- Intrinsic and extrinsic evaluation of clustering results

New Jupyter Notebook!

(relevant for Exercise 4)

Generated by Dall-E 3