WPI

Artificial Intelligence
CS 534

Week 4

Let’s continue

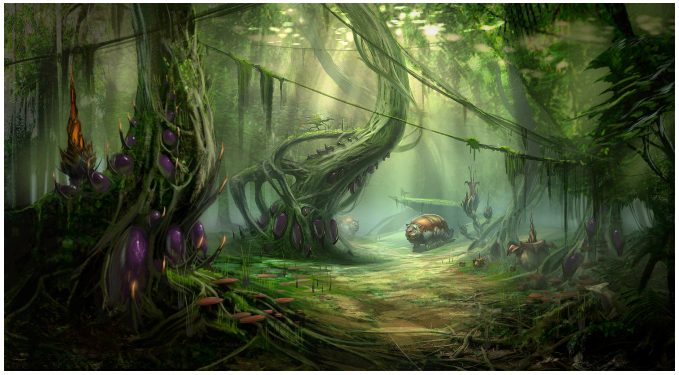
Unit 3: Learning

Materials

- Machine Learning topics:
  - Chapter 18, 20, 21
  - Review papers

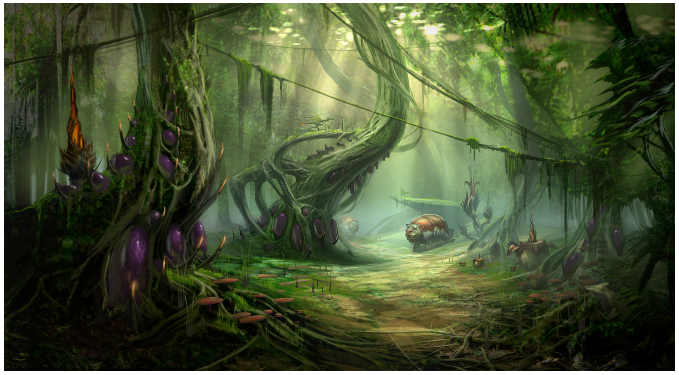
Learning

- Learning is essential for unknown environments
  - *i.e.*, when designer lacks omniscience

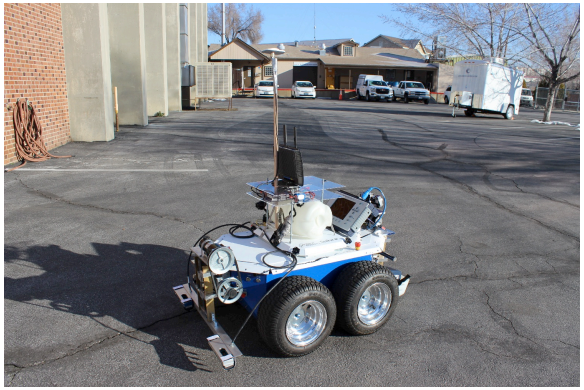


Learning

- Learning is essential for unknown environments
  - *i.e.*, when designer lacks omniscience

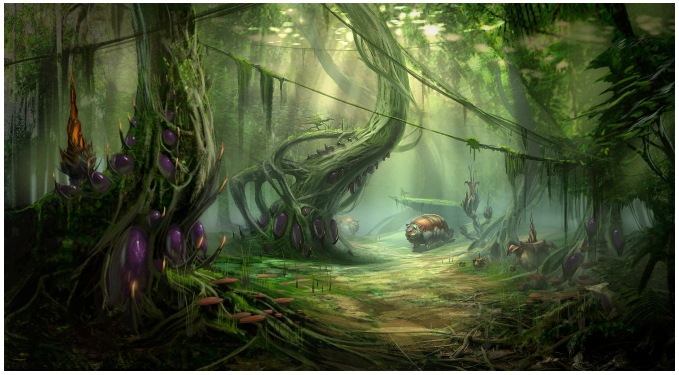


- Learning is useful as a system construction method
  - *i.e.*, expose the agent to reality rather than trying to write it down



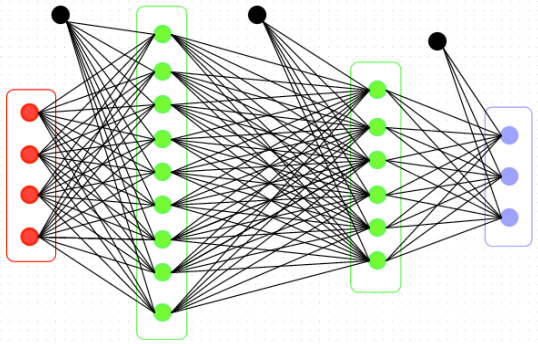
Learning

- Learning is essential for unknown environments
  - *i.e.*, when designer lacks omniscience



- Learning is useful as a system construction method
  - *i.e.*, expose the agent to reality rather than trying to write it down

- Learning modifies the agent's decision mechanisms to improve performance



Feature based approaches in machine learning

Main learning paradigms covered in the class:

- Unsupervised: Clustering
- Supervised: Classification
- Semi-supervised
- Reinforcement learning

Learning element

- Design of a learning element is affected by
  - Which components of the performance element are to be learned
  - What feedback is available to learn these components
  - What representation is used for the components

Learning element

- Design of a learning element is affected by
  - Which components of the performance element are to be learned
  - What feedback is available to learn these components
  - What representation is used for the components

- Type of feedback:
  - Supervised learning: correct answers for each example
  - Unsupervised learning: correct answers not given
  - Semi-supervised learning: some of the correct answers not given
  - Reinforcement learning: occasional rewards

Machine Learning resources

- We will be relying on: scikit-learn — machine learning in Python

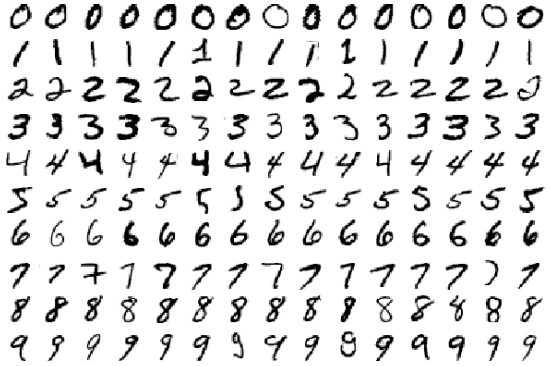
- scikit-learn is a Python module integrating classic machine learning algorithms in the scientific Python world
  - Aims to provide simple and efficient solutions to learning problems, accessible to everybody and reusable in various contexts
  - Free to use
  - Lots of examples and ready to use codes

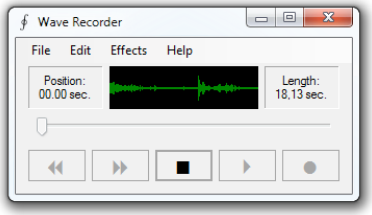
- Unsupervised learning tools:

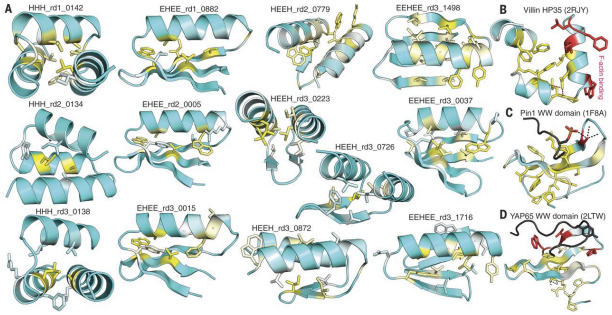
  http://scikit-learn.org/stable/supervised_learning.html

Feature-based representation in learning

Problem: Objects in the real world are diverse and often hard to fully describe

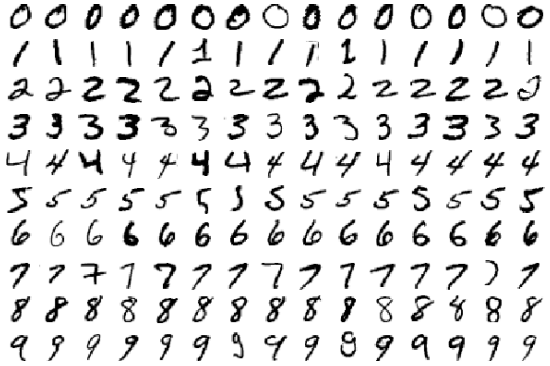


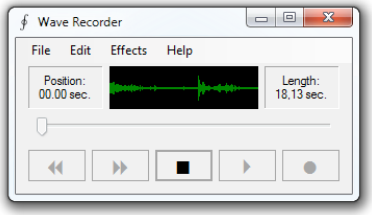


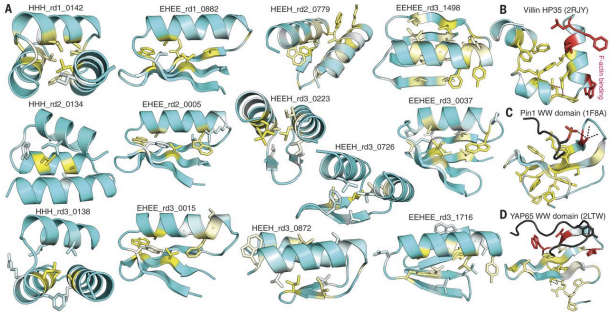


Feature-based representation in learning

Problem: Objects in the real world are diverse and often hard to fully describe

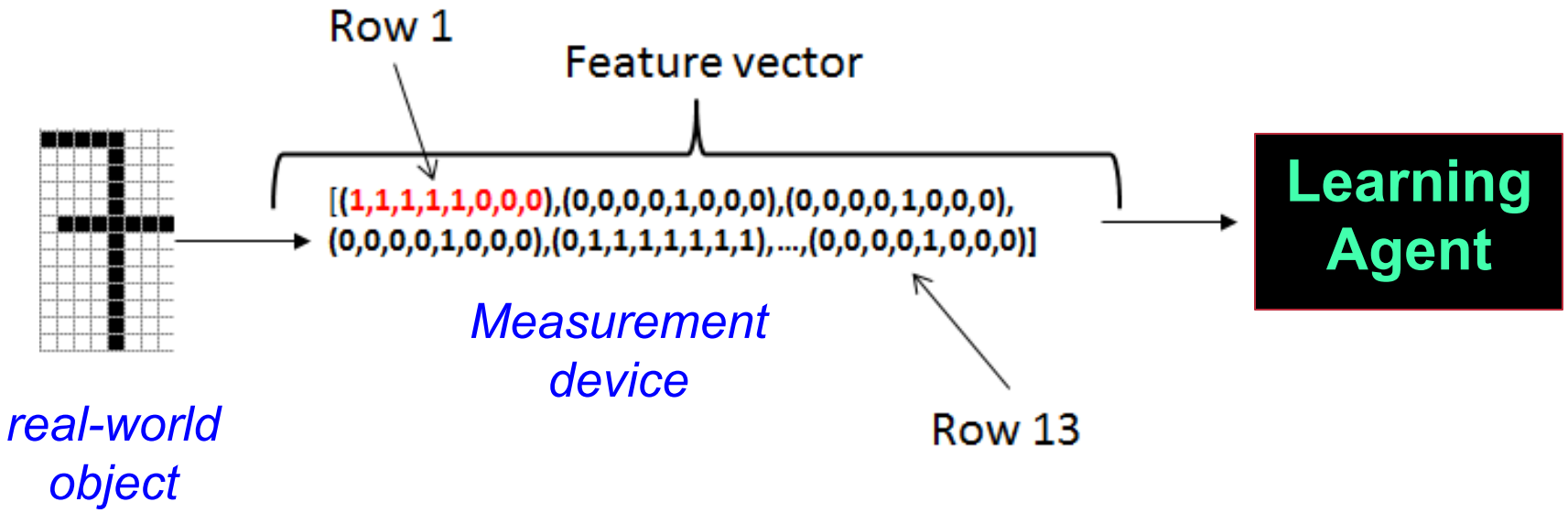






Solution: represent each object as a vector of numerical features

Feature-based representation in learning

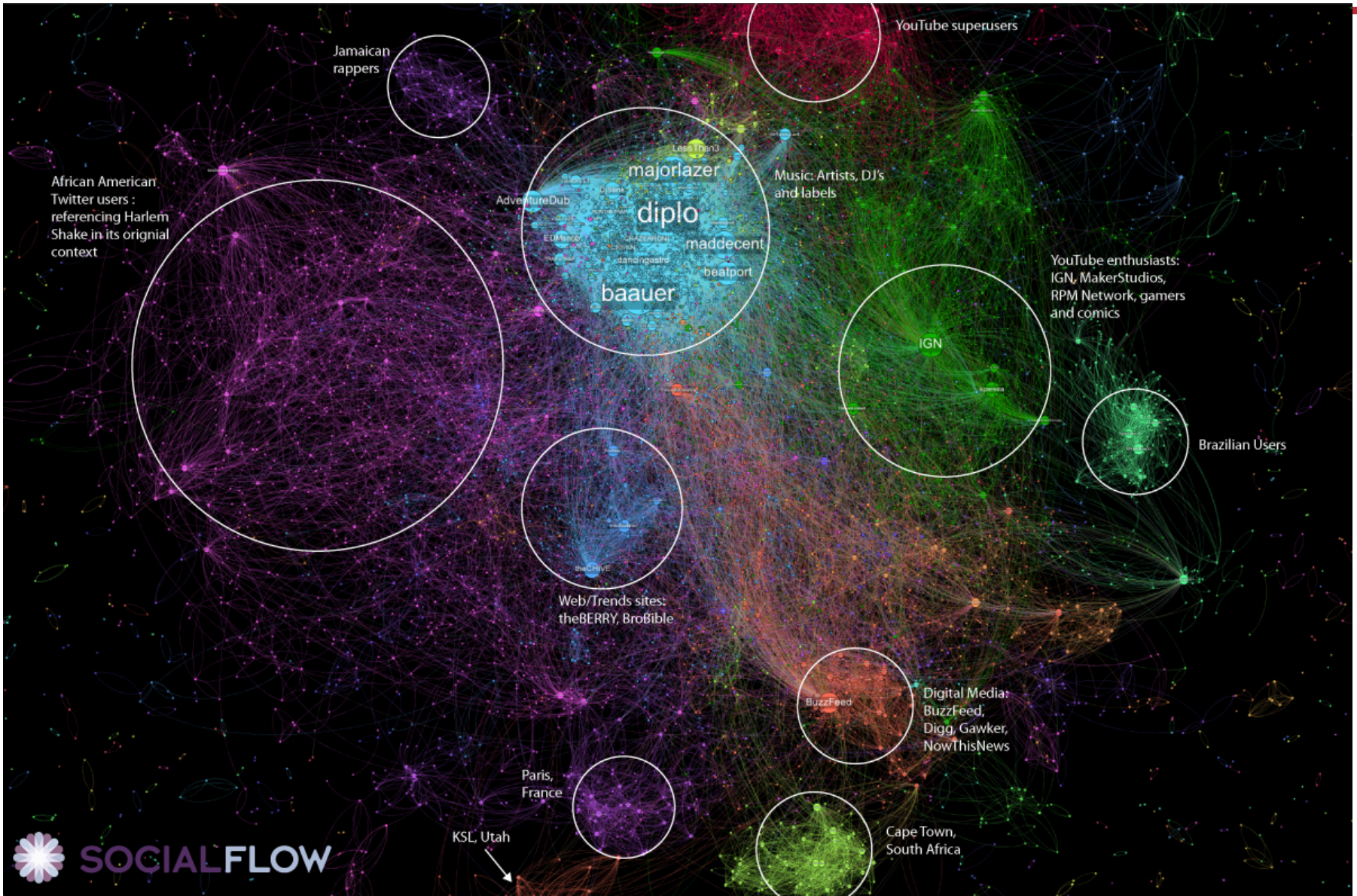


What is unsupervised learning?

- Unsupervised learning is the machine learning task of inferring a function to describe hidden structure from unlabeled data

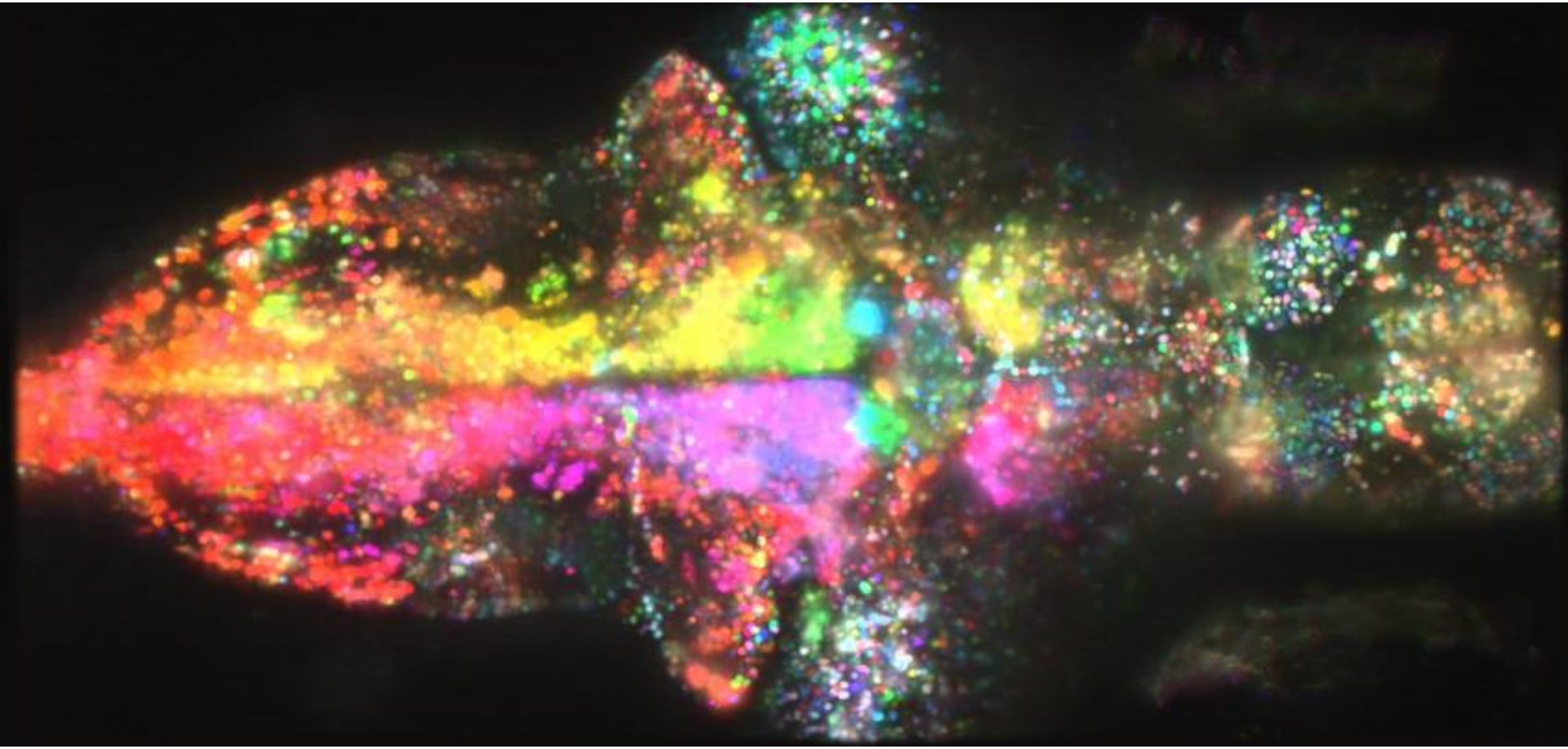
- Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution

Why is it useful?
Example 1: Social network analysis

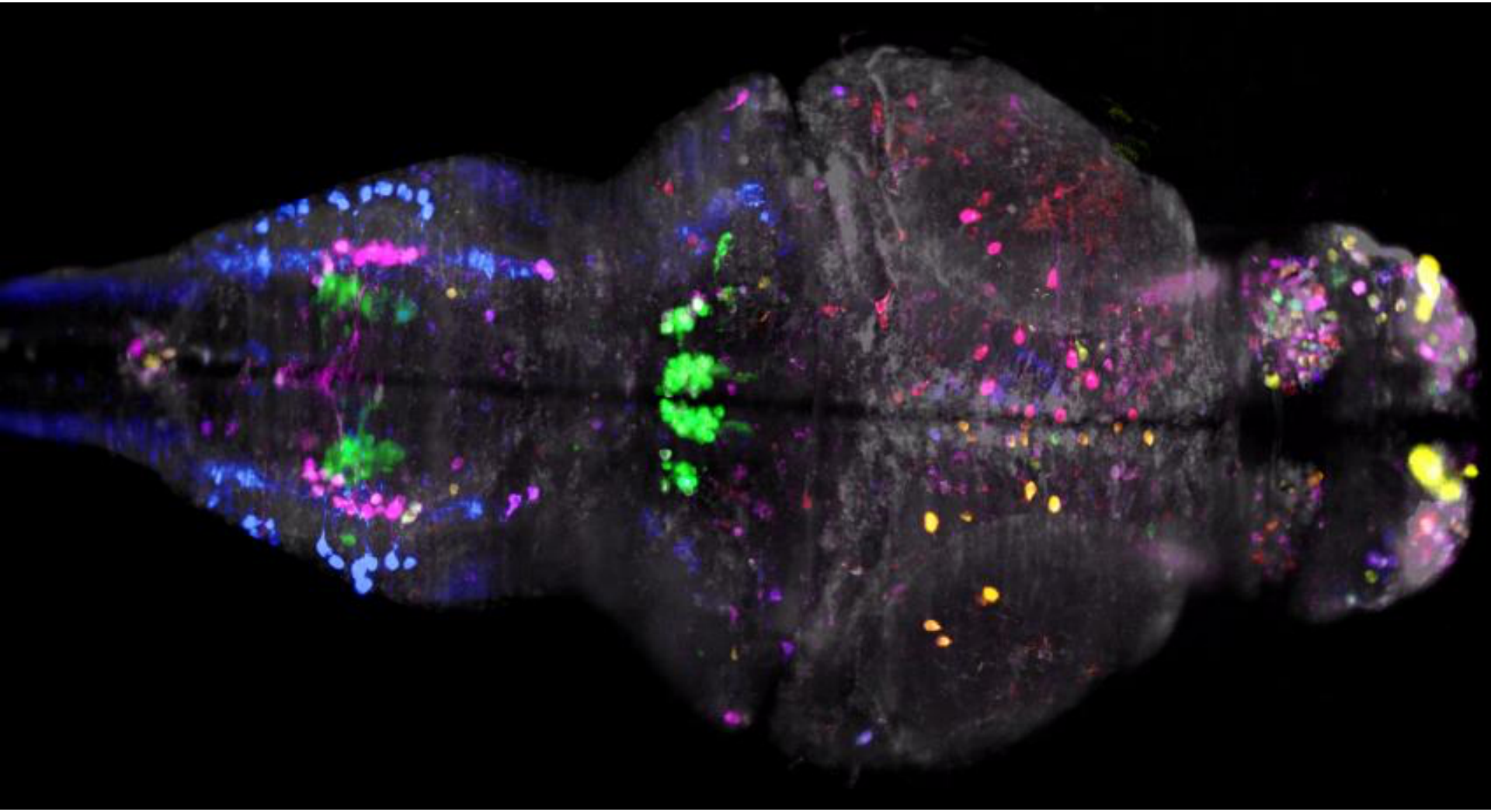


SOCIALFLOW

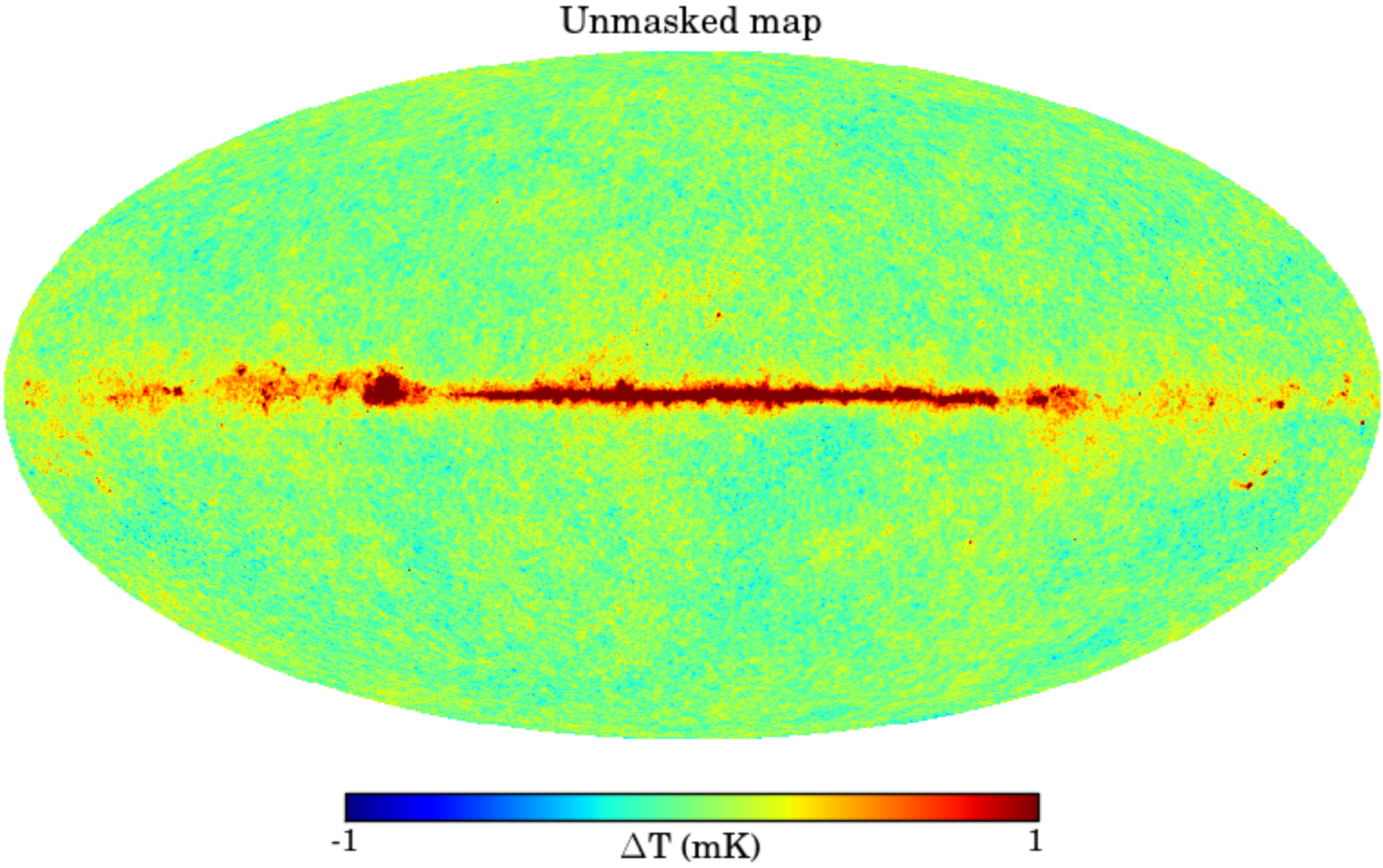
Why is it useful?
Example 2: Brain network



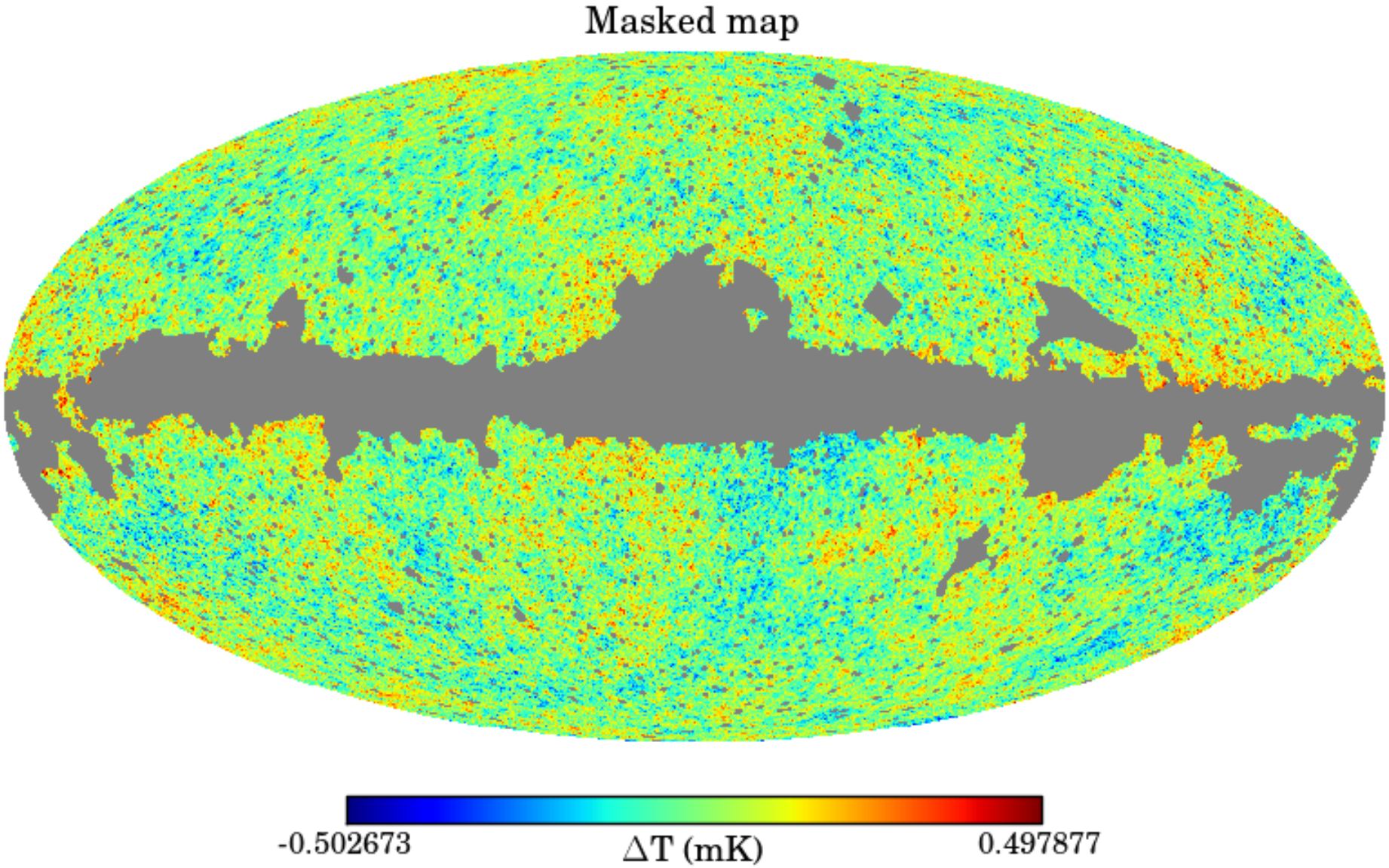
Why is it useful?
Example 2: Brain network



Why is it useful?
Example 3: Astronomical data



Why is it useful?
Example 3: Astronomical data



Clustering goal

In unsupervised learning no training data, with class labeling, are available. The goal becomes: *Group the data into a number of sensible clusters (groups).* This unravels similarities and differences among the available data.

Clustering Applications:

- Engineering
- Bioinformatics
- Social Sciences
- Medicine
- Data and Web Mining

Consider a simple example from zoology

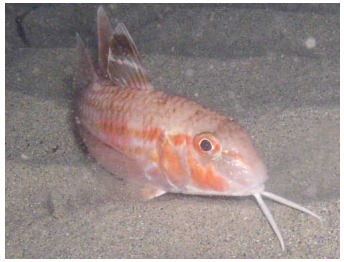
Cluster the following species:

- blue shark
- sheep
- cat
- dog
- lizard
- sparrow
- viper
- seagull
- gold fish
- frog
- red mullet

Consider a simple example from zoology

Cluster the following species:

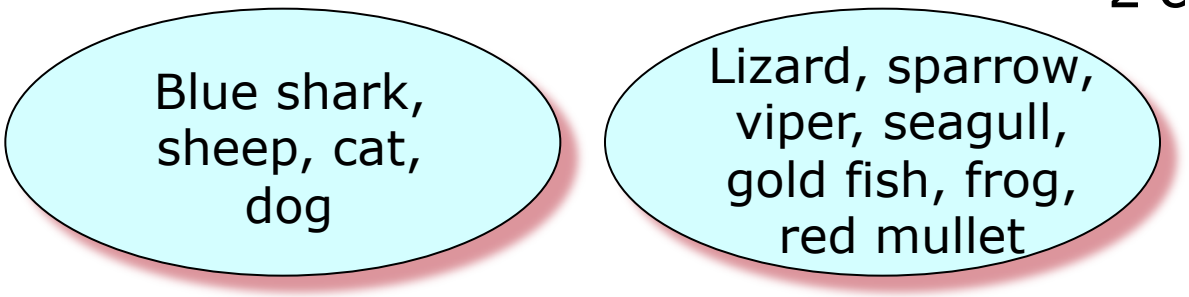
- blue shark
- sheep
- cat
- dog
- lizard
- sparrow
- viper
- seagull
- gold fish
- frog
- red mullet



Two clustering ideas

Idea 1: How mammals bear their progeny

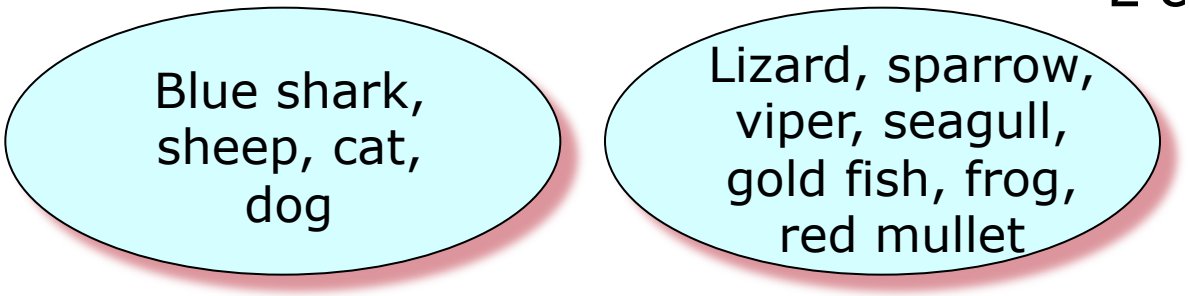
2 clusters



Two clustering ideas

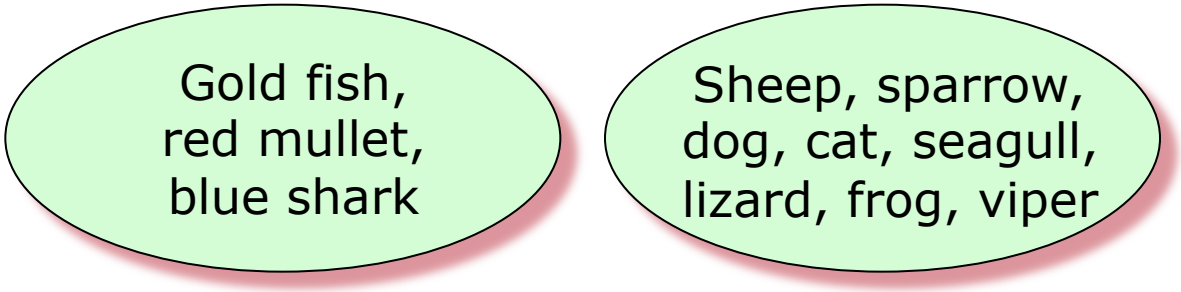
Idea 1: How mammals bear their progeny

2 clusters



Idea 2: Existence of lungs

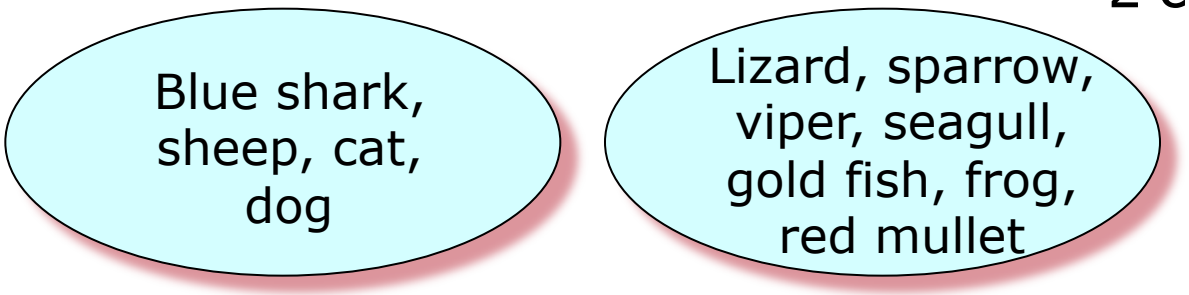
2 clusters



Two clustering ideas

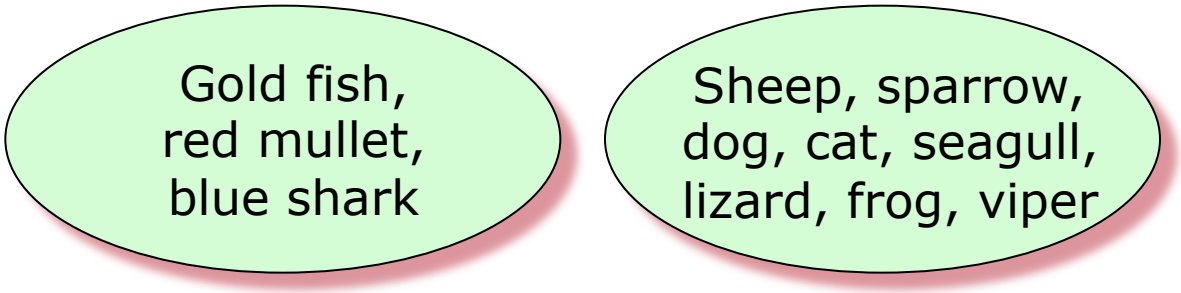
Idea 1: How mammals bear their progeny

2 clusters



Idea 2: Existence of lungs

2 clusters



Conclusion: To perform clustering of a data set, a **clustering criterion** must first be adopted. Different clustering criteria lead, in general, to different clusters.

Basic steps of a clustering task

- Feature Selection: Information rich features-Parsimony (minimum information redundancy)
- Proximity Measure: Quantifies how similar or dissimilar two objects (their representations) are
- Clustering Criterion: Consists of a cost function or some type of rules
- Clustering Algorithm: Consists of the set of steps followed to reveal the structure, based on the similarity measure and the adopted clustering criterion
- Validation of the results
- Interpretation of the results

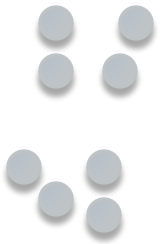
Clustering often means subjectivity

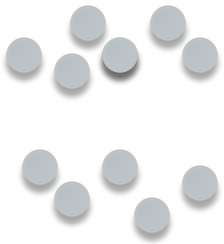
– Depending on the similarity measure, the clustering criterion, and the clustering algorithm—different clusters may result. **Subjectivity** is a reality to live with from now on!

Clustering often means subjectivity

- Depending on the similarity measure, the clustering criterion, and the clustering algorithm—different clusters may result. **Subjectivity** is a reality to live with from now on!

- A simple example: How many clusters??





Clustering often means subjectivity

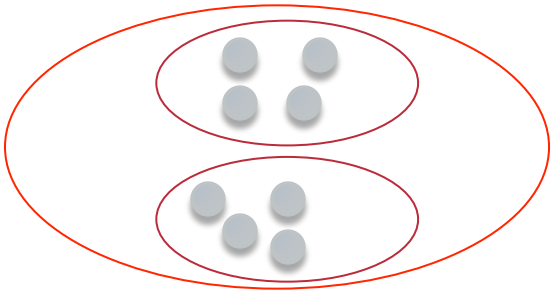
- Depending on the similarity measure, the clustering criterion, and the clustering algorithm—different clusters may result. **Subjectivity** is a reality to live with from now on!

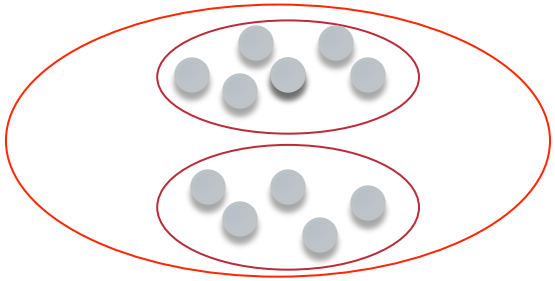
- A simple example: How many clusters??

Clustering often means subjectivity

- Depending on the similarity measure, the clustering criterion, and the clustering algorithm—different clusters may result. **Subjectivity** is a reality to live with from now on!

- A simple example: How many clusters??





2 or 4 ??

Let's move on…

Clustering Basics

Proximity Measures

Basic definitions: Hard clustering

Hard Clustering: Each point belongs to a single cluster

- Let $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$
- An m-clustering of X (notation: $\mathfrak{R}$) is defined as a partition of X into m sets (clusters): $C_1, C_2, \ldots, C_m$, so that

1. $C_i \neq \emptyset, i = 1,2,\ldots,m$
2. $\bigcup_{i=1}^{m} C_i = X$
3. $C_i \cap C_j = \emptyset, \ i \neq j, \ i, j = 1,2,\ldots,m$

In addition: data in C_i are more similar to each other and less similar to the data in the rest of the clusters. Quantifying the terms *similar-dissimilar* depends on the types of clusters that are expected to underlie the structure of X.

Types of features

- With respect to their domain:
  - *Continuous* (the domain is a continuous subset of ***R***).
  - *Discrete* (the domain is a finite discrete set).
    - *Binary* or *dichotomous* (the domain consists of two possible values).

- With respect to the relative significance of the values they take:
  - *Nominal* (the values code states, e.g., the sex of an individual).
  - *Ordinal* (the values are meaningfully ordered, e.g., the rating of the services of a hotel (poor, good, very good, excellent)).
  - *Interval-scaled* (the difference of two values is meaningful but their ratio is meaningless, e.g., temperature).
  - *Ratio-scaled* (the ratio of two values is meaningful, e.g., weight).

Proximity measures: Definitions

Dissimilarity measure (between vectors from X) is a function:

$$d: \; X \times X \longrightarrow \Re$$

with the following properties

- $\exists d_0 \in \Re: \quad -\infty < d_0 \leq d(\boldsymbol{x}, \boldsymbol{y}) < +\infty, \forall \boldsymbol{x}, \boldsymbol{y} \in X$
- $d(\boldsymbol{x}, \boldsymbol{x}) = d_0, \;\; \forall \boldsymbol{x} \in X$
- $d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x}), \forall \boldsymbol{x}, \boldsymbol{y} \in X$

Proximity measures: Definitions (contd.)

If in addition:

- $d(\boldsymbol{x},\boldsymbol{y}) = d_0 \ \textit{if and only if} \ \boldsymbol{x} = \boldsymbol{y}$
- $d(\boldsymbol{x},\boldsymbol{z}) \le d(\boldsymbol{x},\boldsymbol{y}) + d(\boldsymbol{y},\boldsymbol{z}), \ \forall \boldsymbol{x},\boldsymbol{y},\boldsymbol{z} \in X$ (triangular inequality)

(d_0 indicates minimum possible dissimilarity level)

Then d is called a metric dissimilarity measure.

Proximity measures: Definitions (contd.)

– *Similarity measure* (between vectors of X) is a function

$$s: \; X \times X \longrightarrow \Re$$

with the following properties:

- $\exists s_0 \in \Re: \; -\infty < s(\boldsymbol{x}, \boldsymbol{y}) \leq s_0 < +\infty, \;\; \forall \boldsymbol{x}, \boldsymbol{y} \in X$
- $s(\boldsymbol{x}, \boldsymbol{x}) = s_0, \;\; \forall \boldsymbol{x} \in X$
- $s(\boldsymbol{x}, \boldsymbol{y}) = s(\boldsymbol{y}, \boldsymbol{x}), \;\; \forall \boldsymbol{x}, \boldsymbol{y} \in X$

Proximity measures: Definitions (contd.)

If in addition

- $s(\boldsymbol{x}, \boldsymbol{y}) = s_0 \quad if \ and \ only \ if \ \boldsymbol{x} = \boldsymbol{y}$
- $s(\boldsymbol{x}, \boldsymbol{y})s(\boldsymbol{y}, \boldsymbol{z}) \leq [s(\boldsymbol{x}, \boldsymbol{y}) + s(\boldsymbol{y}, \boldsymbol{z})]s(\boldsymbol{x}, \boldsymbol{z}), \ \forall \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in X$

Then s is called a metric similarity measure.

Proximity measures: Definitions (contd.)

Proximity measures between sets:

Let $D_i \subset X$, $i=1,\ldots,k$ and $U=\{D_1,\ldots,D_k\}$

A proximity measure $\wp$ on U is a function

$$\wp : U \times U \longrightarrow \Re$$

Note: A dissimilarity measure has to satisfy the relations of dissimilarity measure between vectors, where D_i's and D_j's are used in place of x, y (similarly for similarity measures)

Dissimilarity between real-valued vectors: l_p measure

- The weighted l_p:

$$d_p(x,y) = \left(\sum_{i=1}^{l} w_i |x_i - y_i|^p \right)^{1/p}$$

- $p = 2$: weighted Euclidean distance

$$d_2(x,y) = \sqrt{(\boldsymbol{x}-\boldsymbol{y})^T B(\boldsymbol{x}-\boldsymbol{y})}$$

- $p = 1$: Manhattan norm

- $p = \infty$: $d_\infty(x,y) = \max_{1 \le i \le l} w_i |x_i - y_i|$

- Other dissimilarity measure $d_G(x,y) = -\log_{10}\left(1 - \frac{1}{l}\sum_{i=1}^{l} \frac{|x_i - y_i|}{|a_i - b_i|}\right)$,

where a_i and b_i are maximum and minimum values for coordinate j

Similarity between real-valued vectors

- Inner product

$$S_{inner}(x,y) = \boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{y} = \sum_{i=1}^{l} x_i y_i \quad \Rightarrow \quad d_{inner}(x,y) = b_{MAX} - S_{inner}(x,y)$$

- Cosine similarity

$$S_{\cos ine}(x,y) = \frac{\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|}$$

- Pearson's correlation coefficient:

$$S_{Pearson}(x,y) = \frac{\boldsymbol{x}_{\boldsymbol{d}}^{\boldsymbol{T}}\boldsymbol{y}_{\boldsymbol{d}}}{\|\boldsymbol{x}_{\boldsymbol{d}}\|\|\boldsymbol{y}_{\boldsymbol{d}}\|}, \boldsymbol{x}_{\boldsymbol{d}} = \left[x_1 - \overline{x}, x_2 - \overline{x}, \ldots, x_l - \overline{x}\right]^T, \boldsymbol{y}_{\boldsymbol{d}} = \left[y_1 - \overline{y}, y_2 - \overline{y}, \ldots, y_l - \overline{y}\right]^T$$

- Tanimoto measure

$$S_T(x,y) = \frac{\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{y}}{\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - \boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{y}}$$

Similarity and dissimilarity between discrete-valued vectors

- Consider vectors whose coordinates belong to a finite set: F=[0,1,…k-1]
- There are exactly k^l vectors of l dimensions
- For vectors ***x***, ***y***, define $A(\boldsymbol{x}, \boldsymbol{y})=[a_{ij}]$, a kxk matrix, called *contingency table*, where a_{ij} is the number of positions where: the first vector has the i symbol and the second has the j symbol (both are from F)

Dissimilarity:

- Hamming distance

$$d_H(x,y)=\sum_{i=o}^{k-1}\sum_{j=0,i\neq j}^{k-1} a_{ij}$$

Similarity:

- Tanimoto measure

$$s_T(x,y)=\frac{\sum_{i=0}^{k-1} a_{ii}}{n_x+n_y-\sum_{i=1}^{k-1}\sum_{j=1}^{k-1} a_{ij}},\quad n_x=\sum_{i=1}^{k-1}\sum_{j=0}^{k-1} a_{ij},\quad n_y=\sum_{i=o}^{k-1}\sum_{j=1}^{k-1} a_{ij}$$

What if we are missing data

For some vectors of the data set X, some features values are unknown

Ways to face the problem:

1. Discard all vectors with missing values (not recommended for small data sets)

2. Find the mean value m_i of the available *i*-th feature values over that data set and substitute the missing *i*-th feature values with m_i

What if we are missing data (contd.)

3. Define b_i=0, if both the *i-th* features x_i, y_i are available and 1 otherwise Then:

$$\wp(\boldsymbol{x},\boldsymbol{y}) = \frac{l}{l - \sum_{i=1}^{l} b_i} \sum_{all\ i:\ b_i=0} \varphi(x_i, y_i)$$

where $\varphi(x_i,y_i)$ denotes the proximity between two scalars x_i, y_i

4. Find the average proximities $\varphi_{avg}(i)$ between all feature vectors in X along all components. Then:

$$\wp(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{l} \psi(x_i, y_i)$$

where $\psi(x_i,y_i)=\varphi(x_i,y_i)$, if both x_i and y_i are available and $\varphi_{avg}(i)$ otherwise

Proximity functions between a vector and a set

- Let $X=\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ and $C \subset X$, $\boldsymbol{x} \in X$

- *All points of C contribute to the definition of* $\wp(\boldsymbol{x}, C)$
  - Max proximity function

$$\wp^{ps}_{\max}(\boldsymbol{x}, C) = \max_{\boldsymbol{y} \in C} \wp(\boldsymbol{x}, \boldsymbol{y})$$

  - Min proximity function

$$\wp^{ps}_{\min}(\boldsymbol{x}, C) = \min_{\boldsymbol{y} \in C} \wp(\boldsymbol{x}, \boldsymbol{y})$$

  - Average proximity function

$$\wp^{ps}_{avg}(\boldsymbol{x}, C) = \frac{1}{n_C} \sum_{\boldsymbol{y} \in C} \wp(\boldsymbol{x}, \boldsymbol{y})$$

(n_C is the cardinality of C)

Point representatives

A representative(s) of C, $\mathbf{r}_C$, contributes to the definition of $\wp(\mathbf{x}, C)$

In this case: $\wp(\mathbf{x}, C) = \wp(\mathbf{x}, \mathbf{r}_C)$

Typical representatives are:

- The mean vector:

$$\mathbf{m}_p = \left(1/n_C\right) \sum_{\mathbf{y} \in C} \mathbf{y}$$

where n_C is the cardinality of C

- The mean center:

$$\mathbf{m}_C \in C: \sum_{\mathbf{y} \in C} d(\mathbf{m}_C, \mathbf{y}) \leq \sum_{\mathbf{y} \in C} d(\mathbf{z}, \mathbf{y}), \ \forall \mathbf{z} \in C$$

d: a dissimilarity measure

- The median center:

$$\mathbf{m}_{med} \in C: \ med(d(\mathbf{m}_{med}, \mathbf{y}) \mid \mathbf{y} \in C) \leq med(d(\mathbf{z}, \mathbf{y}) \mid \mathbf{y} \in C), \ \forall \mathbf{z} \in C$$

NOTE: Other representatives (e.g., hyperplanes, hyperspheres) are useful in certain applications (*e.g.*, object identification using clustering techniques)

Proximity functions between sets

Let $X=\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, $D_i, D_j \subset X$ and $n_i=|D_i|$, $n_j=|D_j|$

All points of each set contribute to $\wp(D_i,D_j)$

1. Max proximity function (measure but not metric, only if $\wp$ is a similarity measure):

$$\wp^{ss}_{\max}(D_i, D_j) = \max_{\boldsymbol{x}\in D_i, \boldsymbol{y}\in D_j} \wp(\boldsymbol{x}, \boldsymbol{y})$$

2. Min proximity function (measure but not metric, only if $\wp$ is a dissimilarity measure):

$$\wp^{ss}_{\min}(D_i, D_j) = \min_{\boldsymbol{x}\in D_i, \boldsymbol{y}\in D_j} \wp(\boldsymbol{x}, \boldsymbol{y})$$

3. Average proximity function (not a measure, even if $\wp$ is a measure):

$$\wp^{ss}_{avg}(D_i, D_j) = \left(1 / n_i n_j \right) \sum_{x\in D_i} \sum_{x\in D_j} \wp(\boldsymbol{x}, \boldsymbol{y})$$

Proximity functions between sets (contd.)

Each set D_i is represented by its representative vector $\boldsymbol{m}_i$

- Mean proximity function (it is a measure provided that $\wp$ is a measure):

$$\wp_{mean}^{ss}(D_i, D_j) = \wp(\boldsymbol{m}_i, \boldsymbol{m}_j)$$

- $$\wp_{e}^{ss}(D_i, D_j) = \sqrt{\frac{n_i n_j}{n_i + n_j}}\wp(\boldsymbol{m}_i, \boldsymbol{m}_j)$$

NOTE: Proximity functions between a vector $\boldsymbol{x}$ and a set C may be derived from the above functions if we set $D_i = \{\boldsymbol{x}\}$.

Part I: Clustering Algorithms

Number of possible clusterings

Let $X=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_N\}$

Question: In how many ways the N points can be assigned into m groups?

Answer:

$$S(N,m)=\frac{1}{m!}\sum_{i=0}^{m}(-1)^{m-1}\binom{m}{i}i^N$$

(so-called Stirling numbers of the second kind)

Examples:

$$S(15,3)=2\,375\,101$$

$$S(20,4)=45\,232\,115\,901$$

$$S(100,5)=10^{68}!!$$

A way out

Consider only a small fraction of clusterings of X and select a "sensible" clustering among them

- Question 1: Which fraction of clusterings is considered?
- Question 2: What "sensible" means?
- The answer depends on the specific clustering algorithm and the specific criteria to be adopted.

Major categories of clustering algorithms

- Sequential: A single clustering is produced. One or few sequential passes on the data

- Hierarchical: A sequence of (nested) clusterings is produced
  - Agglomerative
    - Matrix theory
    - Graph theory
  - Divisive
  - Combinations of the above (*e.g.*, the Chameleon algorithm.)

Sequential clustering algorithms

- The common traits shared by these algorithms are:
  - One or very few passes on the data are required
  - The number of clusters is not known a-priori, except (possibly) an upper bound, q
  - The clusters are defined with the aid of
    - An appropriately defined distance $d(\boldsymbol{x}, C)$ of a point from a cluster
    - A threshold Θ associated with the distance

Basic sequential algorithmic scheme (BSAS)

- $m = 1$ \{number of clusters\}
- $C_m = \{\boldsymbol{x}_1\}$
- For $i=2$ to N
  - Find C_k: $d(\boldsymbol{x}_i, C_k) = min_{1 \le j \le m} d(\boldsymbol{x}_i, C_j)$
  - If $(d(\boldsymbol{x}_i, C_k) > \Theta)$ AND $(m < q)$ then
    - $m = m+1$
    - $C_m = \{\boldsymbol{x}_i\}$
  - Else
    - $C_k = C_k \cup \{\boldsymbol{x}_i\}$
    - *Where necessary, update representatives (*)*
  - End {if}
- End {for}

Basic sequential algorithmic scheme (BSAS)

- $m = 1$ \{number of clusters\}
- $C_m = \{\boldsymbol{x}_1\}$
- For $i=2$ to N
  - Find C_k: $d(\boldsymbol{x}_i, C_k) = min_{1 \le j \le m} d(\boldsymbol{x}_i, C_j)$
  - If $(d(\boldsymbol{x}_i, C_k) > \Theta)$ AND $(m < q)$ then
    - $m = m+1$
    - $C_m = \{\boldsymbol{x}_i\}$
  - Else
    - $C_k = C_k \cup \{\boldsymbol{x}_i\}$
    - *Where necessary, update representatives (*)*
  - End {if}
- End {for}

(*) When the mean vector $\boldsymbol{m}_C$ is used as representative of the cluster C with n_c elements, the updating in the light of a new vector $\boldsymbol{x}$ becomes:

$$\boldsymbol{m}_C^{new} = (n_C \boldsymbol{m}_C + \boldsymbol{x}) / (n_C + 1)$$

Remarks

- The order of presentation of the data in the algorithm plays important role in the clustering results. Different order of presentation may lead to totally different clustering results, in terms of the number of clusters as well as the clusters themselves

- In BSAS the decision for a vector $\boldsymbol{x}$ is reached prior to the final cluster formation

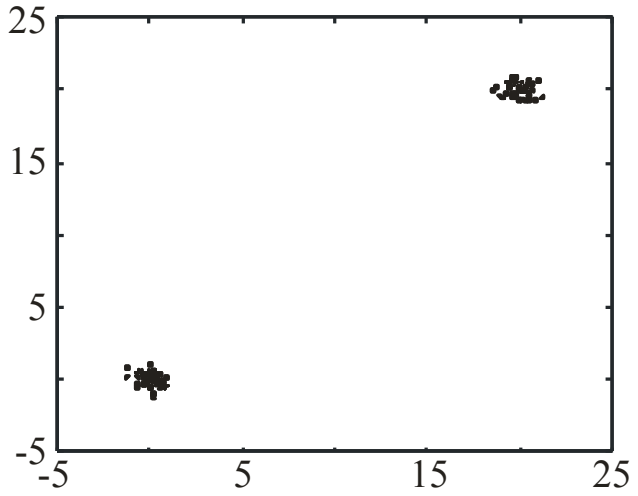
- BSAS perform a single pass on the data. Its complexity is $O(N)$

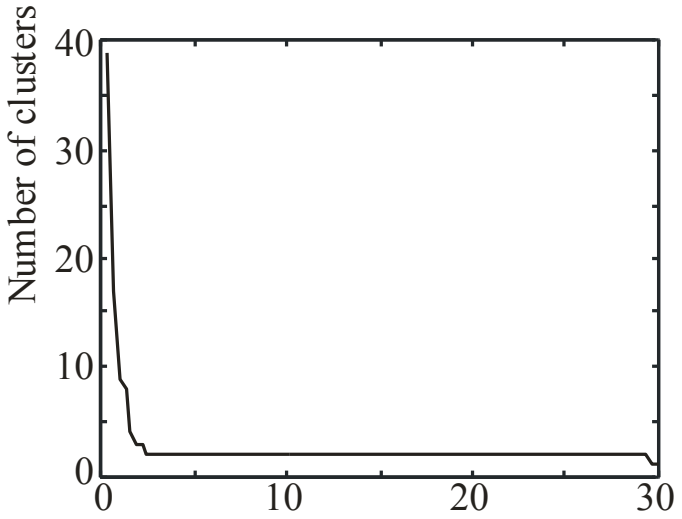
- If clusters are represented by point representatives, compact clusters are favored

Estimating number of clusters in the data set

Let $BSAS(\Theta)$ denote the $BSAS$ algorithm when the dissimilarity threshold is Θ

- For $\Theta = a$ to b step c
  - Run s times $BSAS(\Theta)$, each time presenting the data in a different order
  - Estimate the number of clusters m_Θ, as the most frequent number resulting from the s runs of $BSAS(\Theta)$
- Next Θ
- Plot m_Θ versus Θ and identify the number of clusters m as the one corresponding to the widest flat region in the above graph.





MBSAS, a modification of BSAS

In BSAS, a decision for a data vector $\boldsymbol{x}$ is reached prior to the final cluster formation, which is determined after all vectors have been presented to the algorithm

- MBSAS deals with the above drawback, at the cost of presenting the data twice to the algorithm
- MBSAS consists of:
  - A cluster determination phase (first pass on the data),

    which is the same as BSAS with the exception that no vector is assigned to an already formed cluster. At the end of this phase, each cluster consists of a single element.
  - A pattern classification phase (second pass on the data),

    where each one of the unassigned vector is assigned to its closest cluster.

➢ Remarks:

- In MBSAS, a decision for a vector $\boldsymbol{x}$ during the pattern classification phase is reached taking into account all clusters
- MBSAS is sensitive to the order of presentation of the vectors
- MBSAS requires two passes on the data. Its complexity is $O(N)$

A two-threshold sequential scheme (TTSAS)

- The formation of the clusters, as well as the assignment of vectors to clusters, is carried out concurrently (like **BSAS** and unlike **MBSAS**)
- Two thresholds Θ_1 and Θ_2 ($\Theta_1<\Theta_2$) are employed
- The general idea is the following:
  - If the distance $d(\boldsymbol{x},C)$ of $\boldsymbol{x}$ from its closest cluster, C, is greater than Θ_2 then:
    - *A new cluster represented by* $\boldsymbol{x}$ *is formed.*
  - Else if $d(\boldsymbol{x},C)<\Theta_1$ then
    - $\boldsymbol{x}$ *is assigned to* C.
  - Else
    - *The decision is postponed to a later stage.*
  - End {if}

The unassigned vectors are presented iteratively to the algorithm until all of them are classified

Remarks

- In practice, a few passes (≥ 2) of the data set are required

- TTSAS is less sensitive to the order of data presentation, compared to BSAS

Refinement stages

The problem of closeness of clusters: "*In all the above algorithms it may happen that two formed clusters lie very close to each other*".

A simple merging procedure:

- (*A*) Find C_i, C_j ($i<j$) such that
  $$d(C_i,C_j)=min_{k,r=1,\ldots,m,k\neq r}d(C_k,C_r)$$
- If $d(C_i,C_j)\leq M_1$ then
  \{M_1 is a user-defined threshold\}
  - *Merge C_i, C_j to C_i and eliminate C_j*
  - *If necessary, update the cluster representative of C_i*
  - *Rename the clusters $C_{j+1},\ldots,C_m$ to $C_j,\ldots,C_{m-1}$, respectively*
  - *m=m-1*
  - *Go to (A)*
- Else
  - *Stop*
- End {if}

The problem of sensitivity to the order of data presentation

- The problem of sensitivity to the order of data presentation:

  "A vector x may have been assigned to a cluster C_i at the current stage but another cluster C_j may be formed at a later stage that lies closer to x"

A simple reassignment procedure

- For i=1 to N
  - *Find C_j such that $d(\mathbf{x}_i, C_j)=\min_{k=1,\ldots,m} d(\mathbf{x}_i, C_k)$*
  - *Set b(i)=j \{ b(i) is the index of cluster that lies closest to $\mathbf{x}_i$ \}*
- End {for}
- For j=1 to m
  - *Set $C_j=\{\mathbf{x}_i \in X: b(i)=j\}$*
  - *If necessary, update representatives*
- End {for}

Hierarchical clustering algorithms

- Produce a hierarchy of (**hard**) clusterings instead of a single clustering

- Applications in:
  - Social sciences
  - Biological taxonomy
  - Modern biology
  - Medicine
  - Archaeology
  - Computer science and engineering

Definition

Let $X=\{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N\}$, $\boldsymbol{x}_i=[x_{i1},\ldots,x_{il}]^T$. Recall that:

- In hard clustering each vector belongs exclusively to a single cluster.
- An m-(hard) clustering of X, K, is a partition of X into m sets (clusters) $C_1,\ldots,C_m$, so that:
  - $C_i \neq \varnothing, i=1,2,\ldots,m$
  - $\bigcup_{i=1}^{m} C_i = X$
  - $C_i \cap C = \varnothing,\ i \neq j,\ i,j=1,2,\ldots,m$

  By the definition: $K=\{C_j, j=1,\ldots m\}$

- Definition: A clustering K_1 containing k clusters is said to be nested in the clustering K_2 containing r ($<k$) clusters, if each cluster in K_1 is a subset of a cluster in K_2.

  We write $K_1 \subset K_2$

- Example: Let $K_1=\{\{x_1, x_3\}, \{x_4\}, \{x_2, x_5\}\}$, $K_2=\{\{x_1, x_3, x_4\}, \{x_2, x_5\}\}$,

  $K_3=\{\{x_1, x_4\}, \{x_3\}, \{x_2, x_5\}\}$, $K_4=\{\{x_1, x_2, x_4\}, \{x_3, x_5\}\}$.

  It is $K_1 \subset K_2$, but not $K_1 \subset K_3$, $K_1 \subset K_4$, $K_1 \subset K_1$.

- Remarks:
  - Hierarchical clustering algorithms produce a hierarchy of nested clusterings
  - They involve N steps at the most.
  - At each step t, the clustering K_t is produced by K_{t-1}.

- Main categories:
  - Agglomerative clustering algorithms: Here $K_0=\{\{x_1\}, \ldots, \{x_N\}\}$, $K_{N-1}=\{\{x_1, \ldots, x_N\}\}$ and $K_0 \subset \ldots \subset K_{N-1}$.
  - Divisive clustering algorithms: Here $K_0=\{\{x_1, \ldots, x_N\}\}$, $K_{N-1}=\{\{x_1\}, \ldots, \{x_N\}\}$ and $K_{N-1} \subset \ldots \subset K_0$.

Agglomerative Algorithms

- Let $g(C_i,C_j)$ a proximity function between two clusters of X.

Generalized Agglomerative Scheme (GAS)

- Initialization
  - Choose $K_0=\{\{\boldsymbol{x}_1\},\ldots,\{\boldsymbol{x}_N\}\}$
  - $t=0$
- Repeat
  - $t=t+1$
  - Choose (C_i,C_j) in K_{t-1} such that

$$g(C_i,C_j)=\begin{cases}\min_{r,s}\, g(C_r,C_s), & \text{if } g \text{ is a disim. function}\\ \max_{r,s}\, g(C_r,C_s), & \text{if } g \text{ is a sim. function}\end{cases}$$

  - Define $C_q=C_i \cup C_j$ and produce $K_t=(K_{t-1}-\{C_i,C_j\})\cup\{C_q\}$
- Until all vectors lie in a single cluster.