Pattern Recognition: Clustering

Badri Narayan Subudhi
Indian Institute of Technology Jammu
NH44, Nagrota, Jagti, Jammu
Subudhi.badri@gmail.com

Module no.	Topic	No. of hours
1.	**Bayes Decision Theory:** Minimum-error-rate classification, Classifiers, Discriminant functions, Decision surfaces, Normal density and discriminant functions, discrete features	6
2.	**Parameter Estimation Methods:** Maximum-Likelihood estimation: Gaussian case; Maximum a Posteriori estimation; Bayesian estimation: Gaussian case	6
3.	**Unsupervised learning and clustering:** Criterion functions for clustering; Algorithms for clustering: K-Means, Hierarchical and other methods; Cluster validation; Gaussian mixture models; Expectation-Maximization method for parameter estimation; Maximum entropy estimation	8
4.	**Nonparametric techniques for density estimation:** Parzen-window method; K-Nearest Neighbour method	6
5.	**Dimensionality reduction:** Fisher discriminant analysis; Principal component analysis;	6
6.	**Linear discriminant functions:** Gradient descent procedures; Perceptron; Support vector machines	5
7.	**Non-metric methods for pattern classification :** Non-numeric data or nominal data. Decision trees: Classification and Regression Trees (CART)	5
	Total Lecture hours	**42**

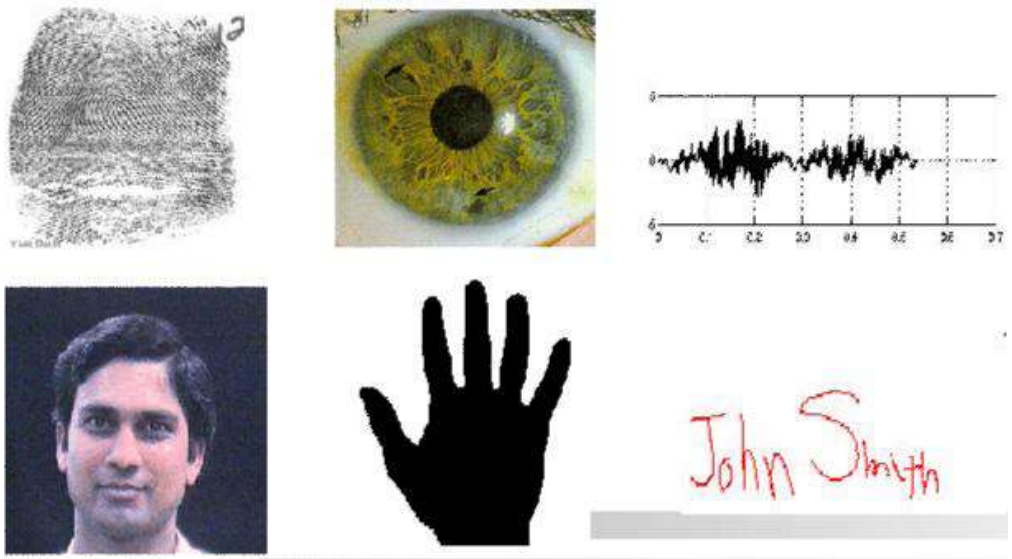
What is Pattern Recognition?

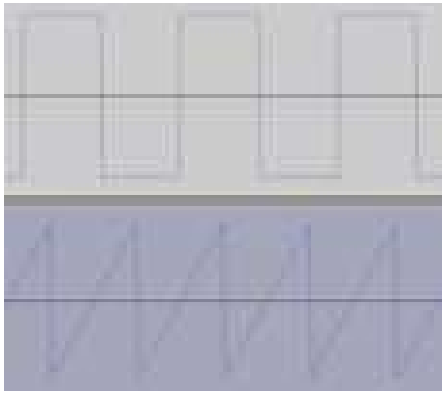
- Pattern Recognition is the study of how machines can:
  - observe the environment,
  - learn to distinguish patterns of interest,
  - make sound and reasonable decisions about the categories of the patterns.

- What is a *pattern*?
- What kinds of *category* we have?

What is a pattern?

- A **pattern** is an abstraction, represented by a set of measurements describing a “physical” object.
- Many types of patterns exist:
  - visual, temporal, sonic, logical, ...

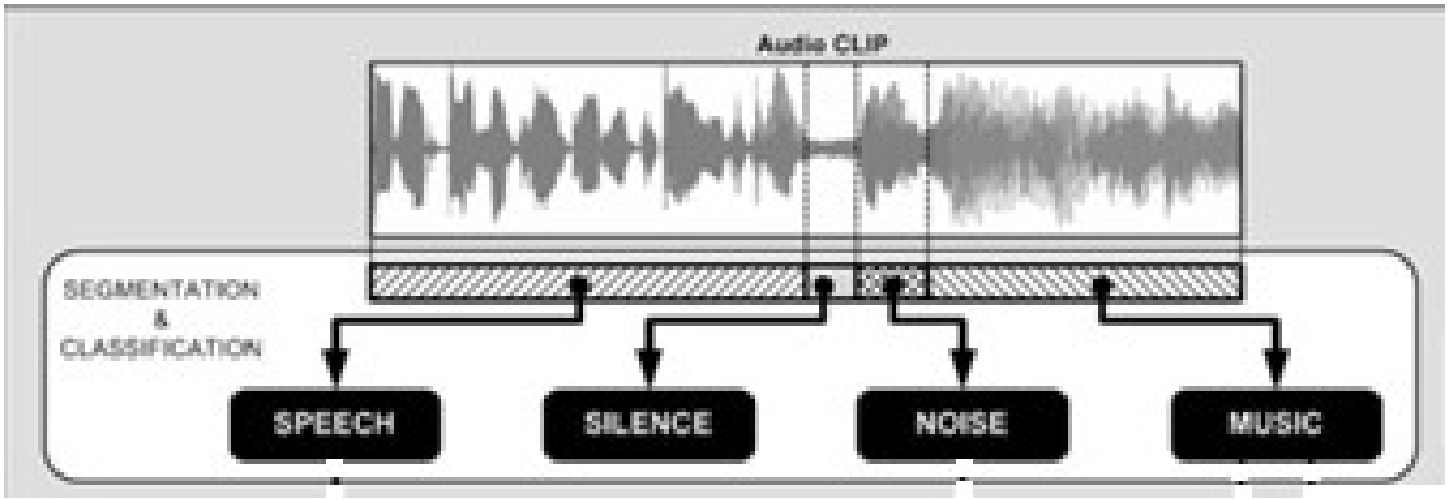




What is a Pattern Class (or category)?

– Is a set of patterns sharing common attributes

– A collection of “similar”, not necessarily identical, objects



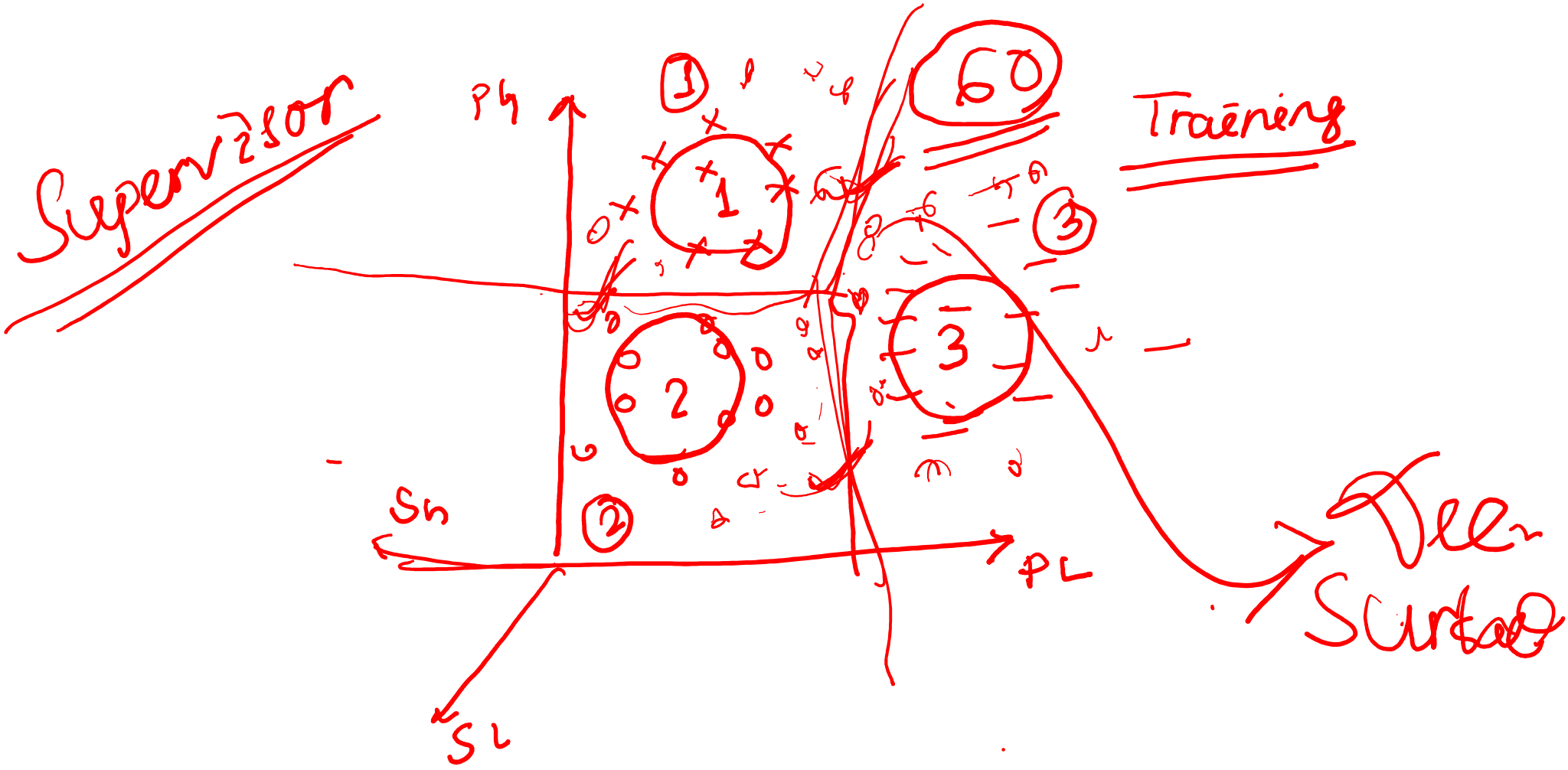
What is **Features** ?

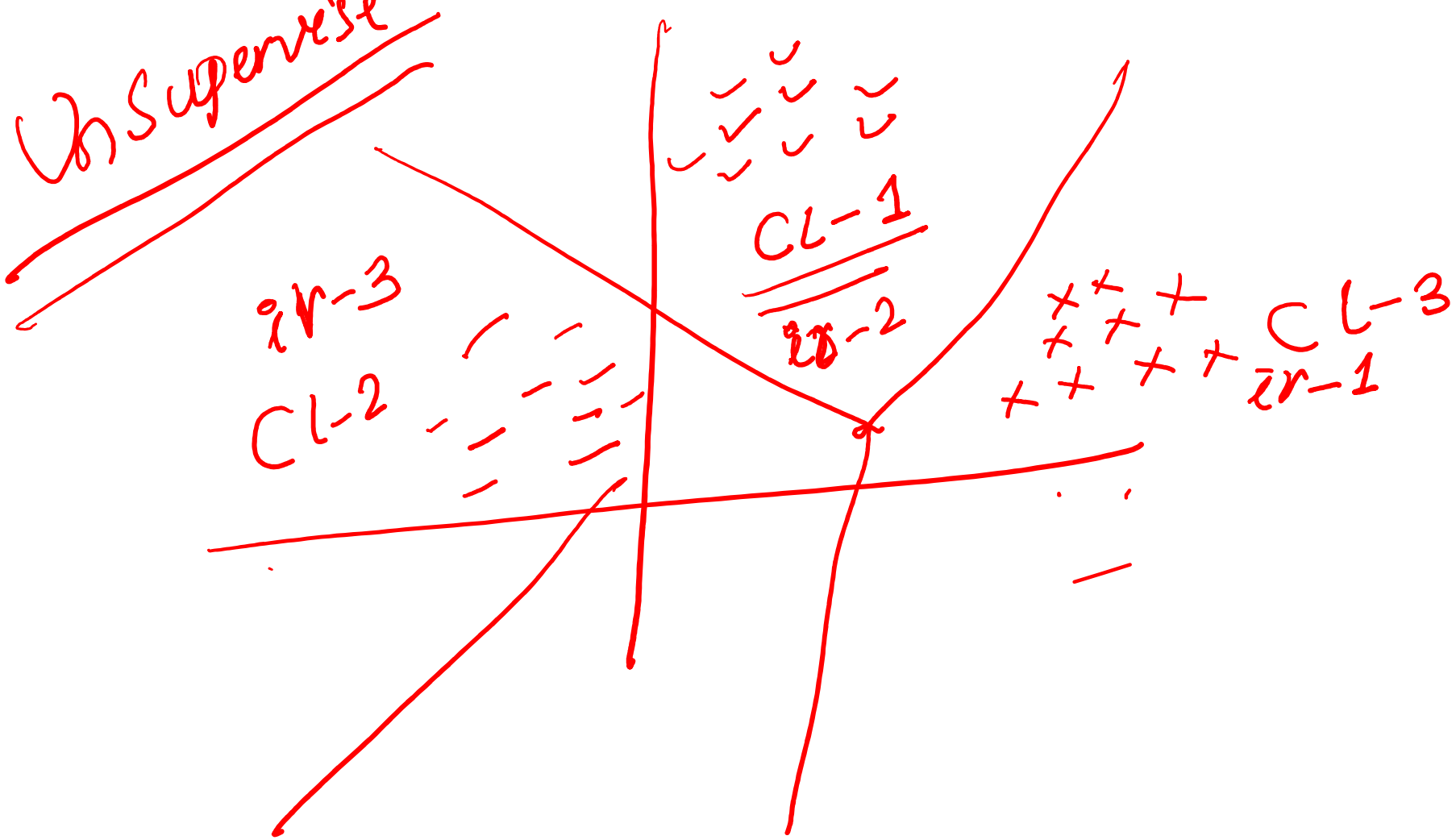
- Features are properties of an object:
  - Ideally representative of a specific type (i.e. class) of objects
  - Perceptual relevant

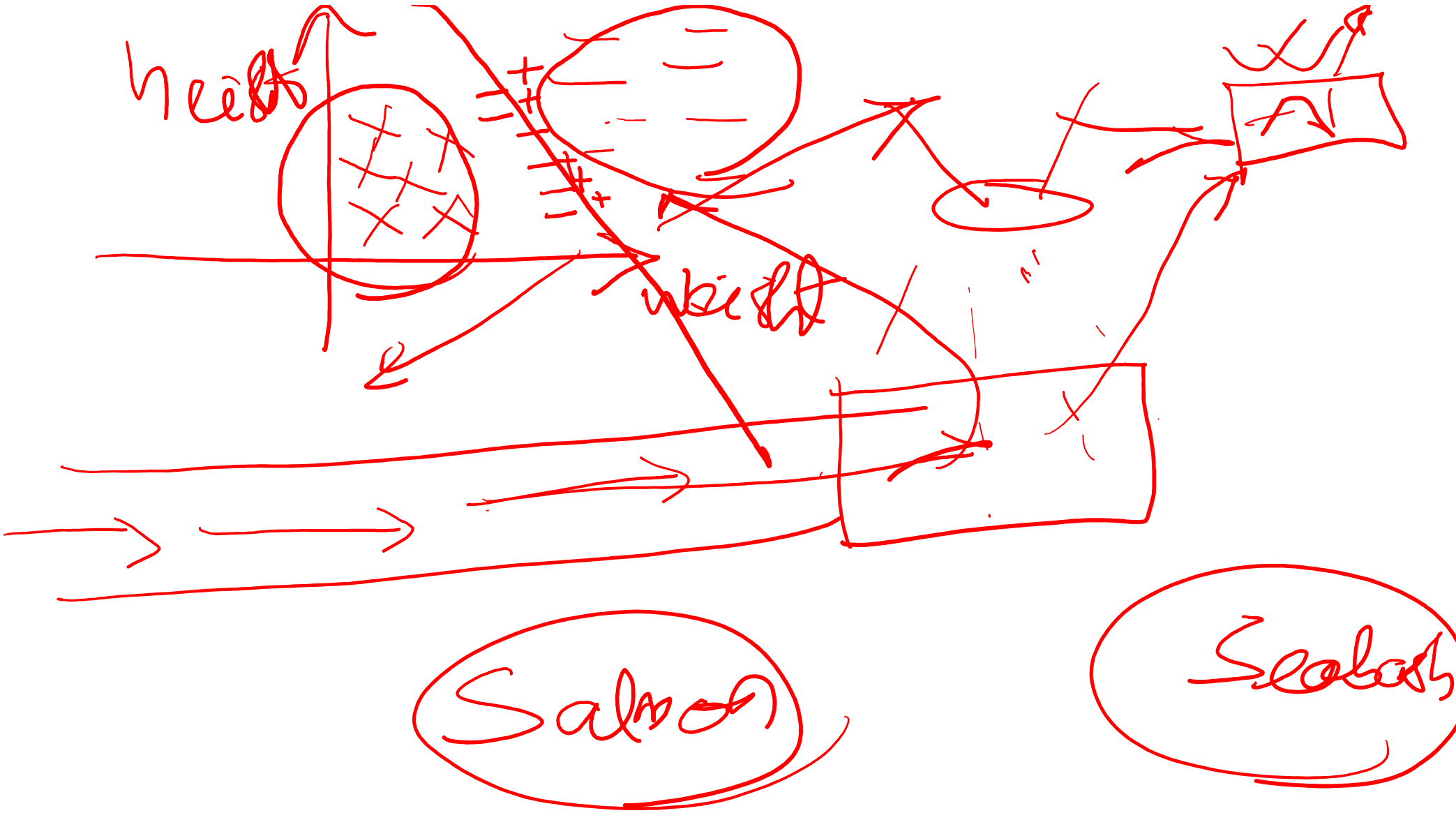
Supervised learning vs. unsupervised learning

- Supervised learning: discover patterns in the data that relate data attributes with a target (class) attribute.
  - These patterns are then utilized to predict the values of the target attribute in future data instances.
- Unsupervised learning: The data have no target attribute.
  - We want to explore the data to find some intrinsic structures in them.

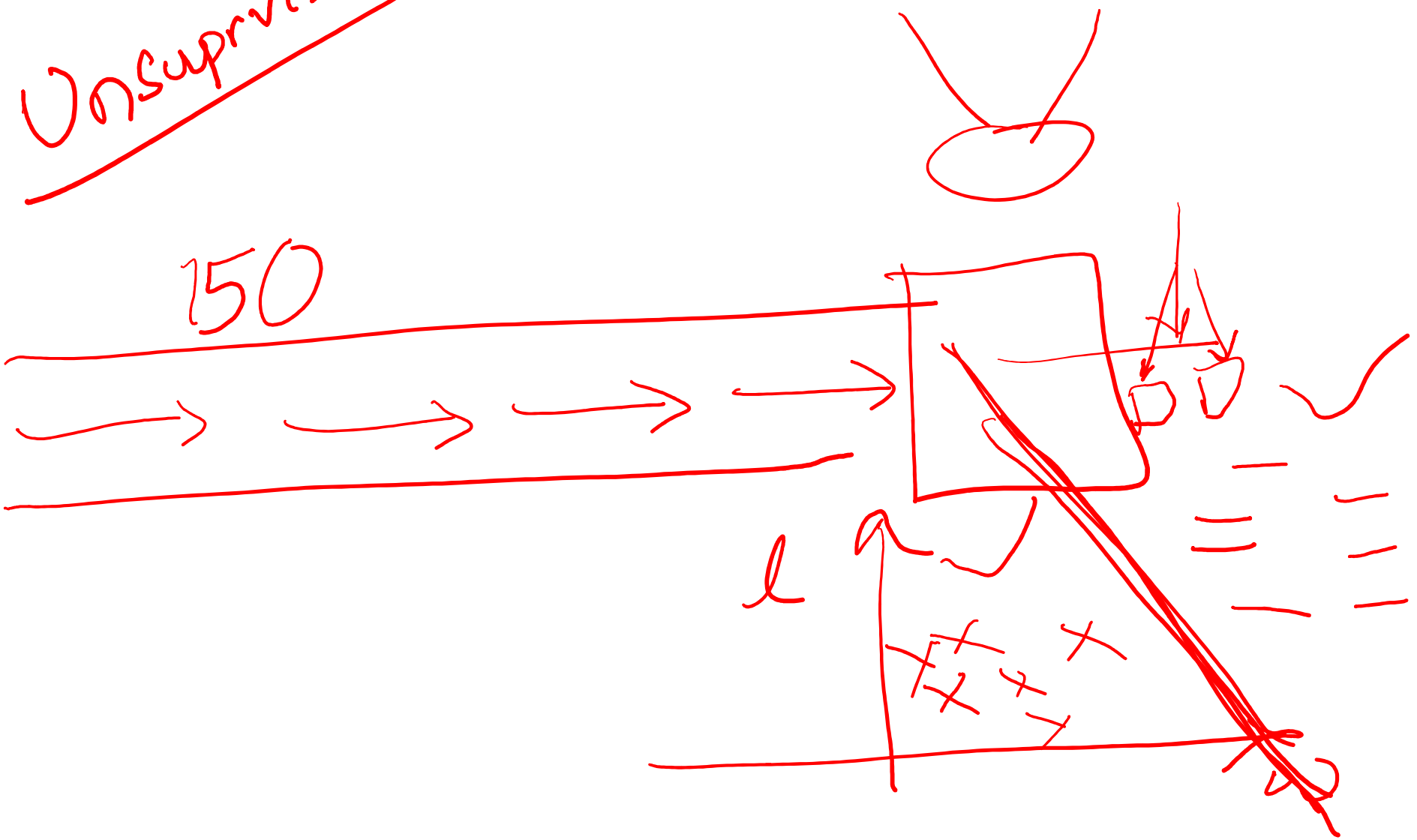

Iris
PL
Pb
SL
Sb
CL
1
2
3
4
5
50
100
150
20
1
20
2
20
3
90


Supervisor
Training
Ph
Sh
PL
SL
1
2
3
60
Test Surface


Supervise
CL-1
ir-2
ir-3
CL-2
CL-3
ir-1


Salmon
Seabass

Unsupervised

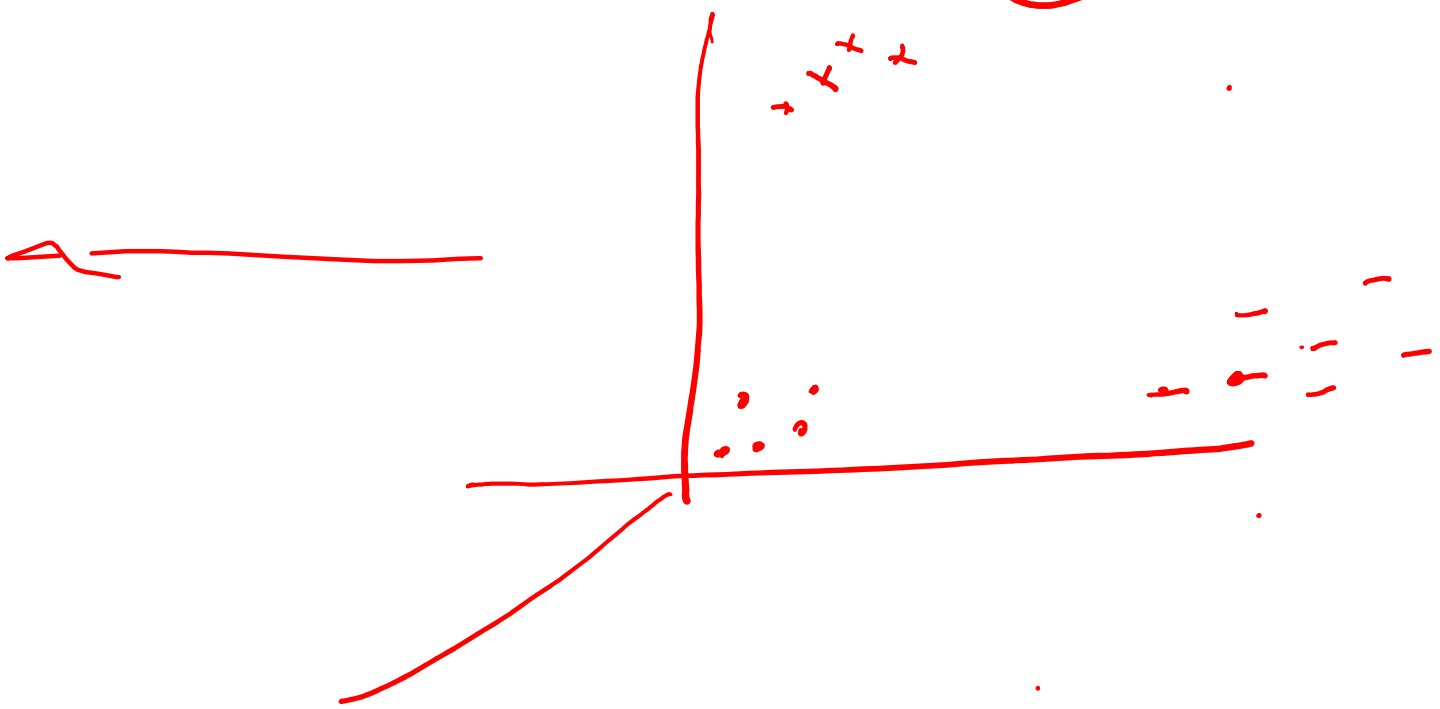


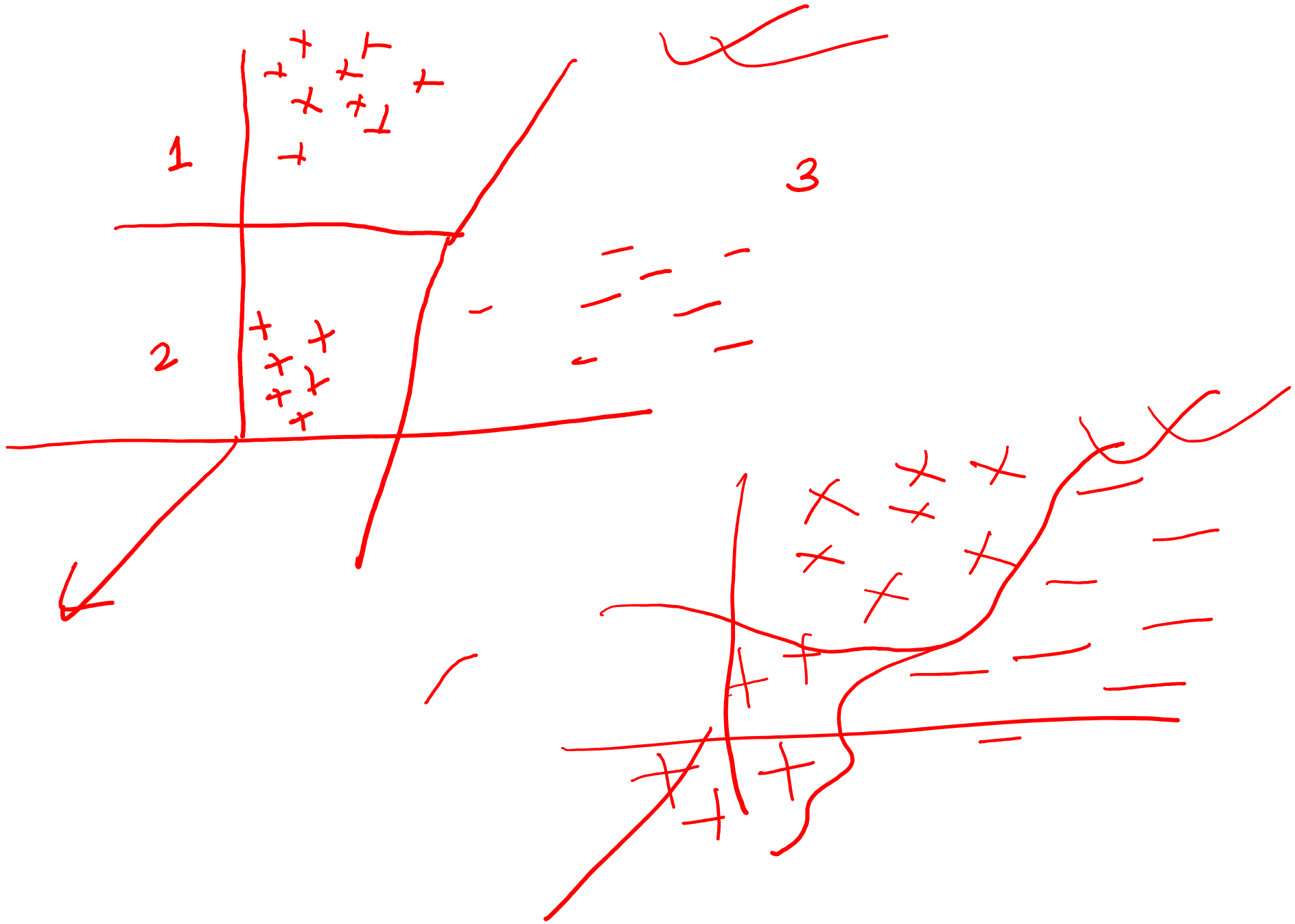
Semi-supervised

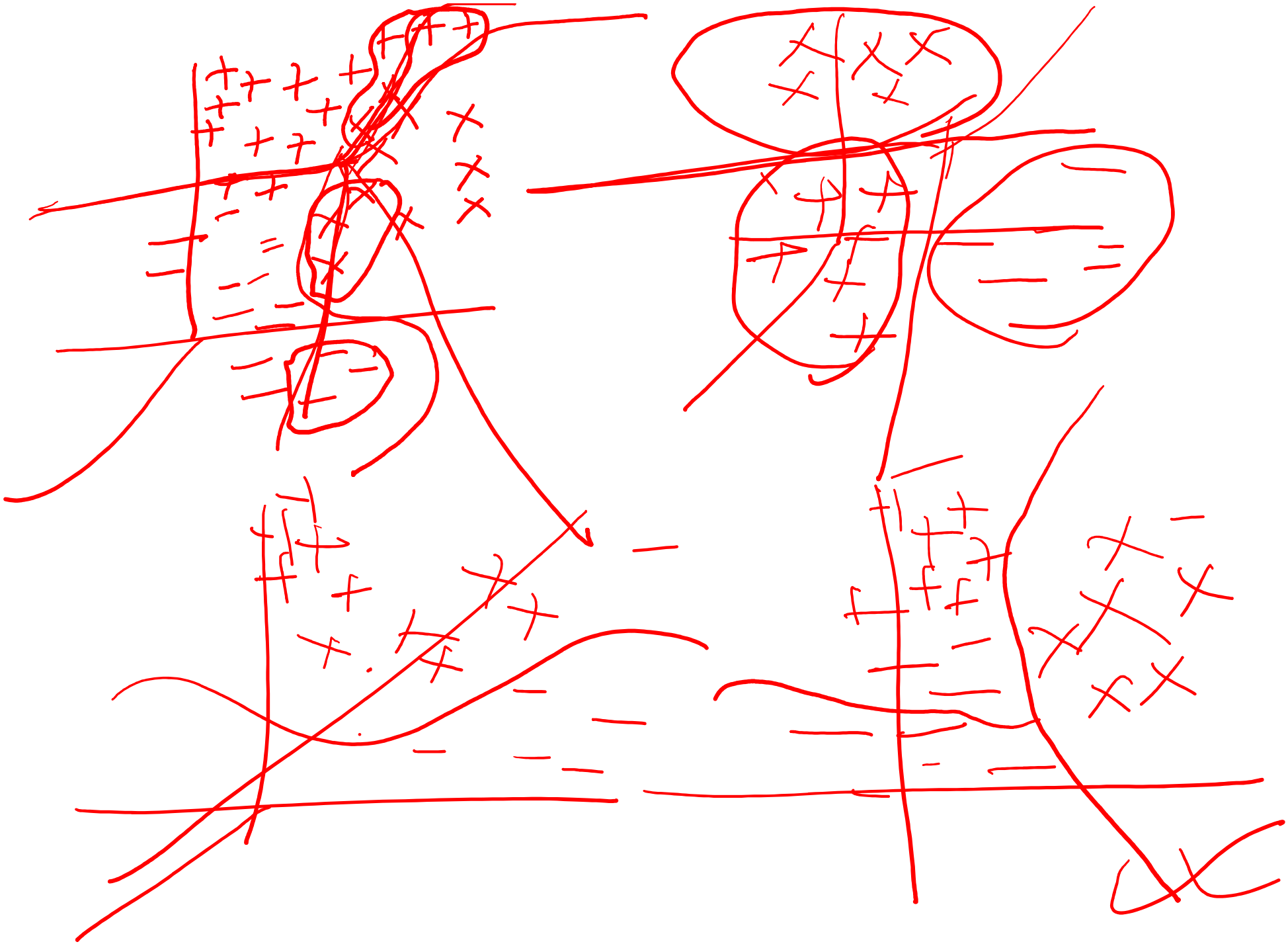
150

60

15



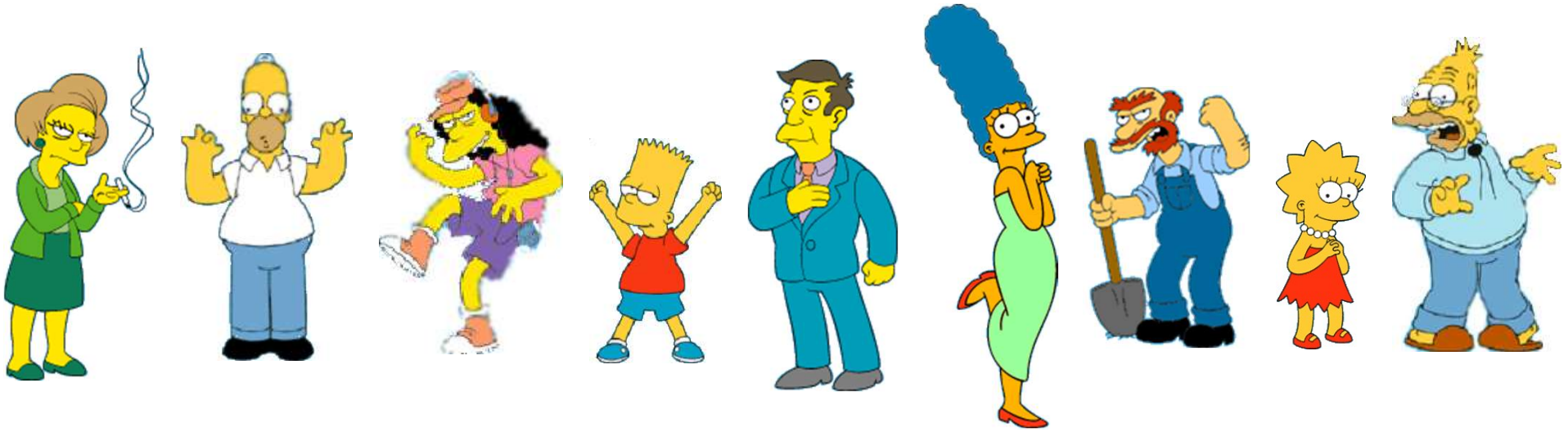

1
2
3



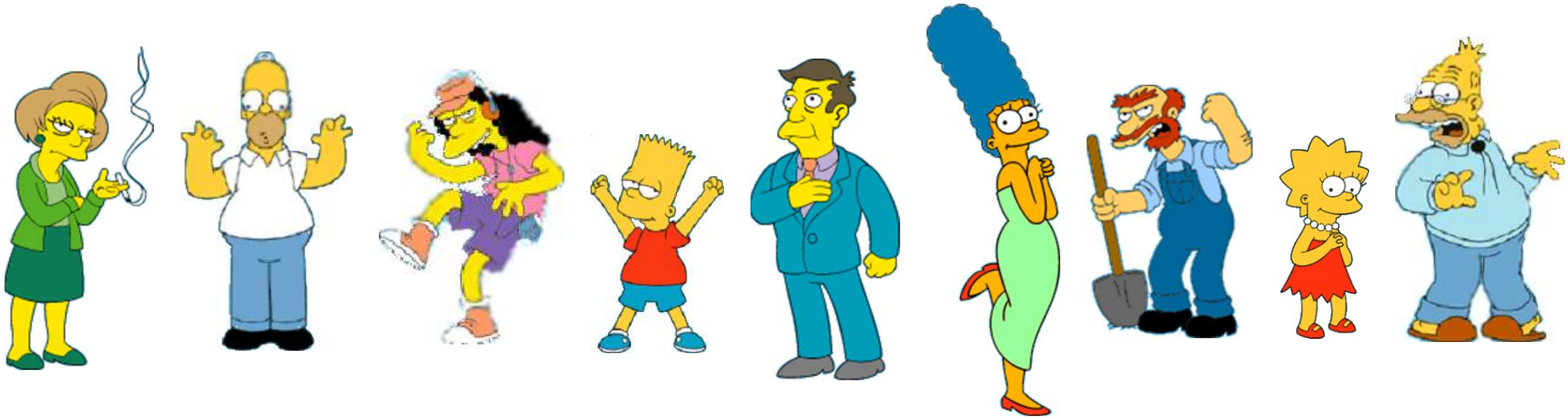
Clustering

- Clustering is a technique for finding similarity groups in data, called **clusters**. I.e.,
  - it groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.
- Clustering is often called an **unsupervised learning** task as no class values denoting an *a priori* grouping of the data instances are given, which is the case in supervised learning.
- Due to historical reasons, clustering is often considered synonymous with unsupervised learning.

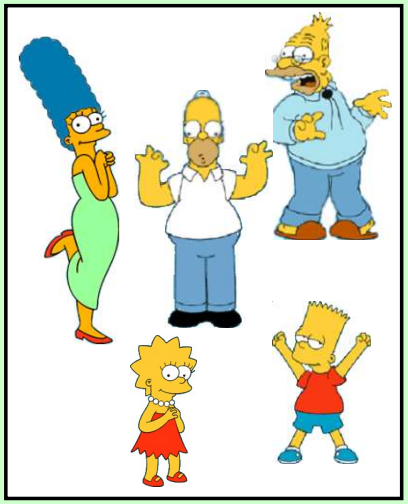
What is a natural grouping of these objects?



What is a natural grouping of these objects?

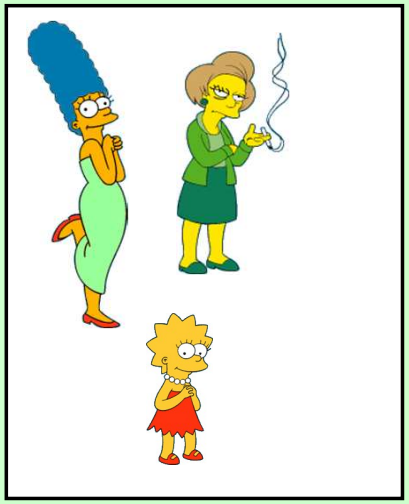


Clustering is subjective



Simpson's Family

School Employees

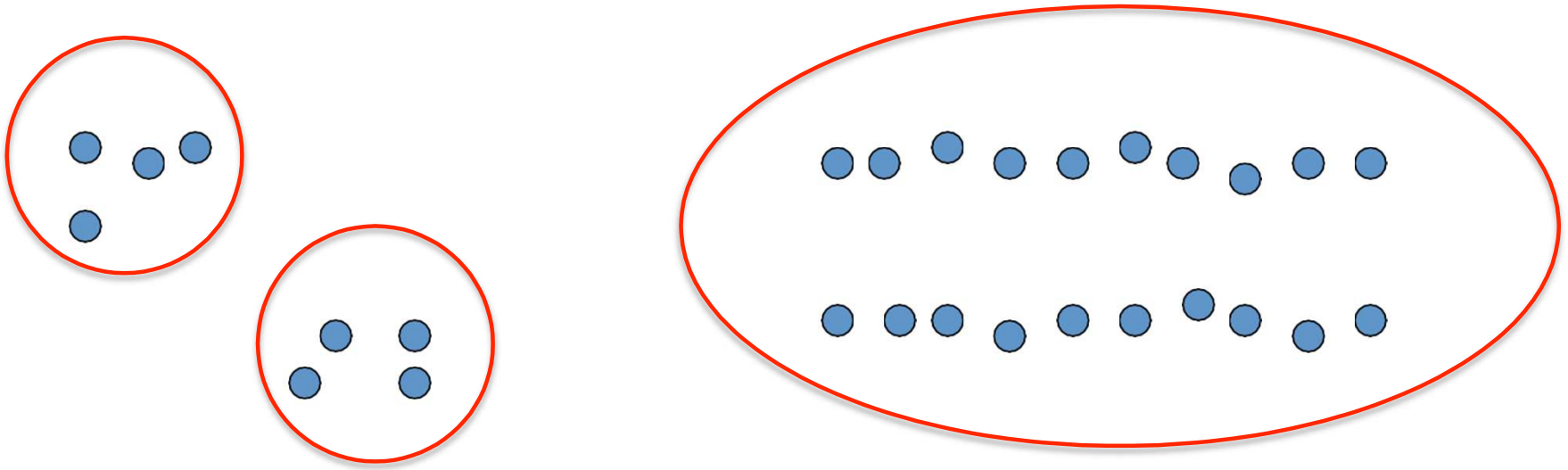


Females

Males

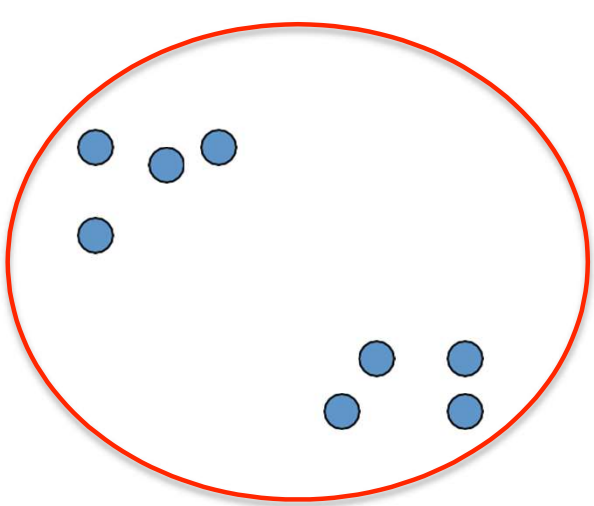
Clustering

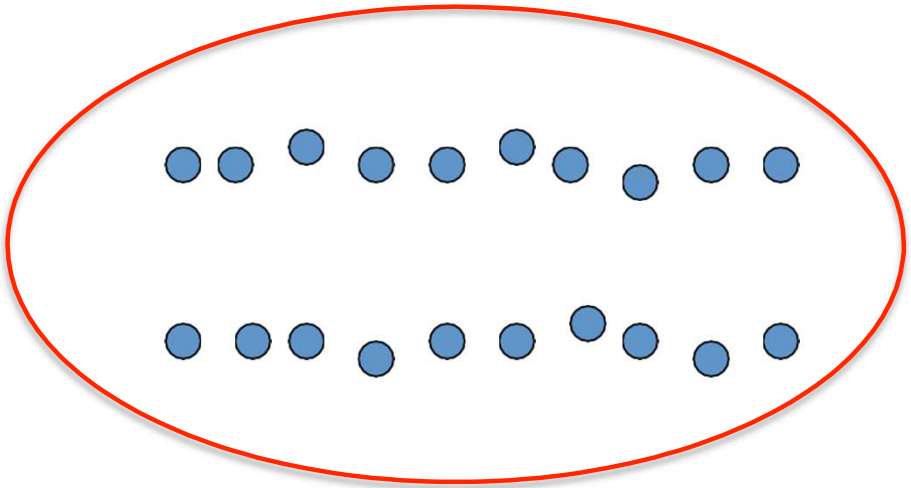
- Basic idea: group together similar instances
- Example: 2D point patterns



Clustering

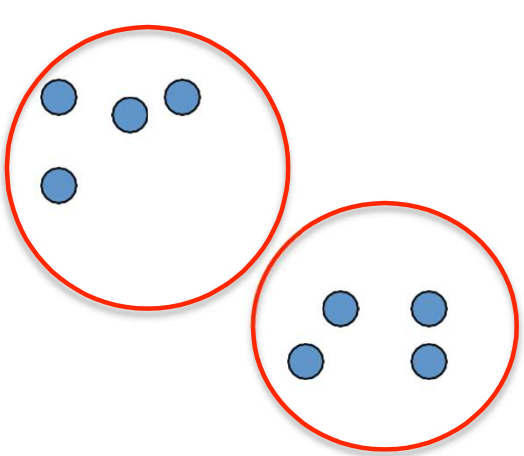
- Basic idea: group together similar instances
- Example: 2D point patterns

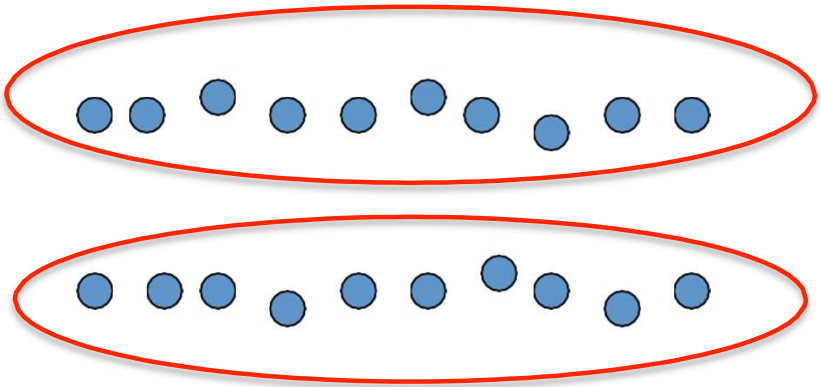




Clustering

- Basic idea: group together similar instances
- Example: 2D point patterns





- What could "similar" mean?
  - One option: small Euclidean distance (squared)

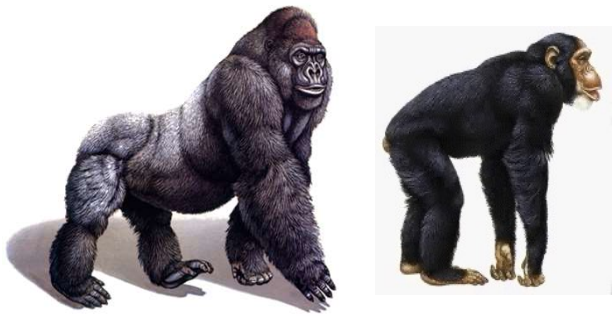
$$\text{dist}(x, y) = ||x - y||_2^2$$

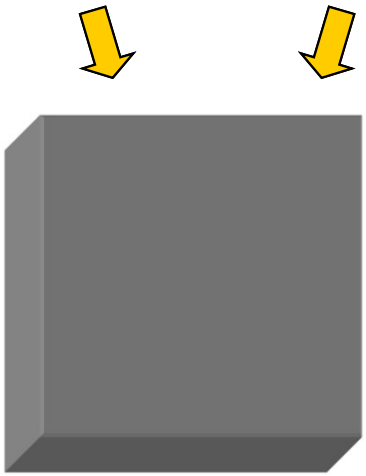
  - Clustering results are crucially dependent on the measure of similarity (or distance) between "points" to be clustered

What is Similarity?

Defining Distance Measures

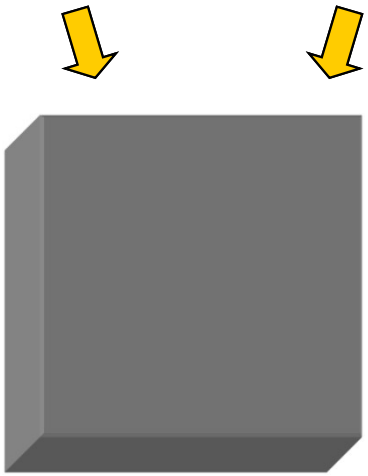
Definition: Let O_1 and O_2 be two objects from the universe of possible objects. The distance (dissimilarity) between O_1 and O_2 is a real number denoted by $D(O_1,O_2)$



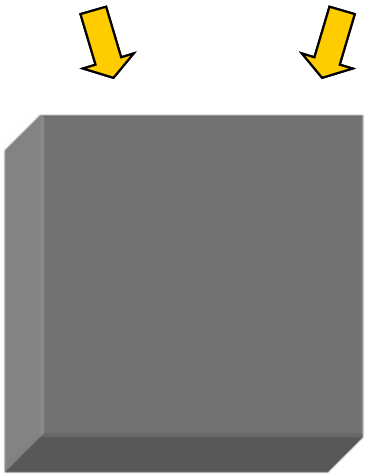


0.23

Peter Piotr



3



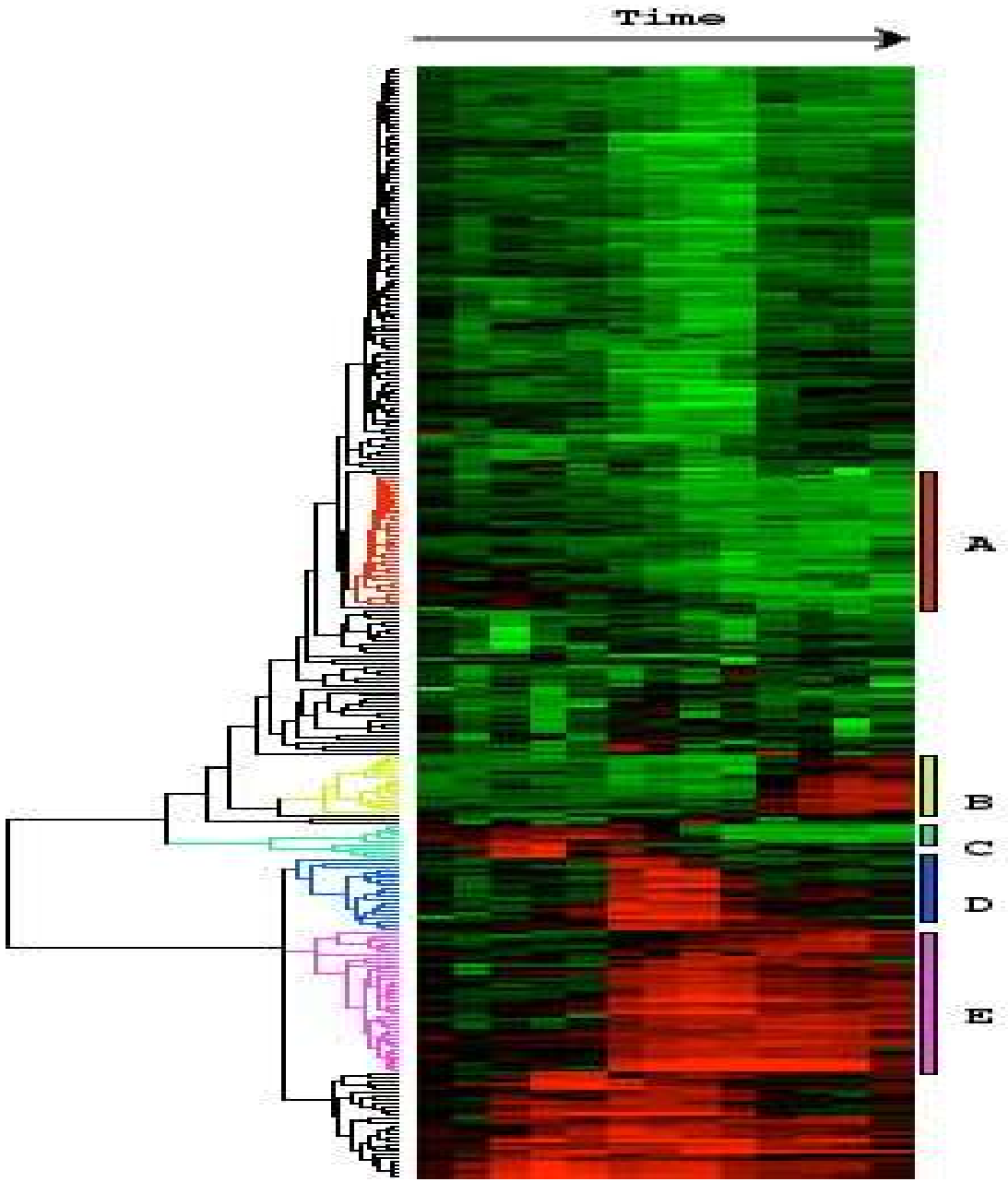
342.7

Clustering examples

Image segmentation

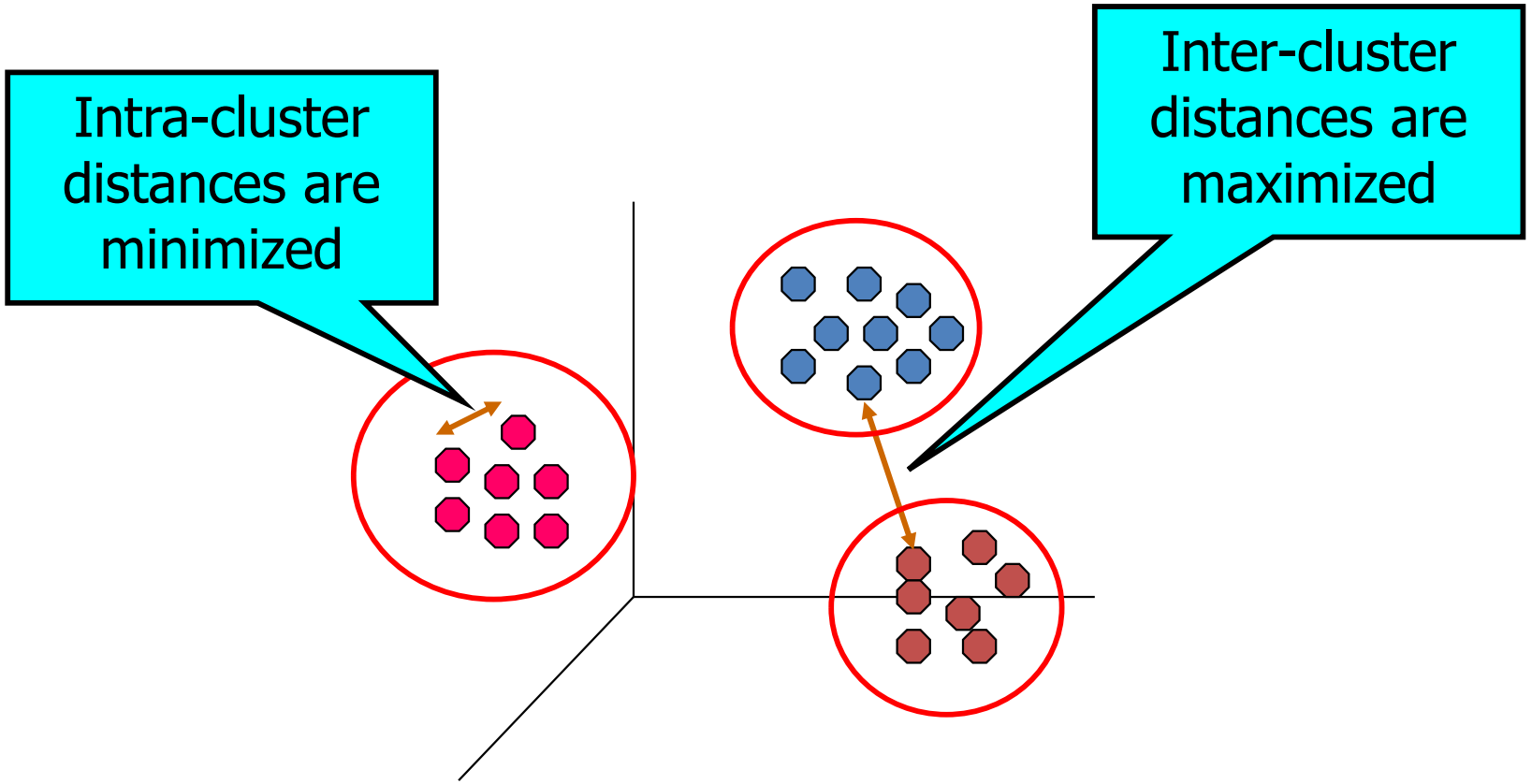
Goal: Break up the image into meaningful or perceptually similar regions

Clustering gene expression data



What is Cluster Analysis?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups



Different Clustering

- Partitioning Clustering: K-mean, K-medoid, PAM
- Fuzzy Clustering: FCM
- Hierarchical Clustering: AGNES, DIANA
- Density-Based Clustering.: DBSCAN, Mean-shift
- OPTICS (Ordering Points to Identify Clustering Structure)
- Kernelized Clustering
- Probabilistic clustering

K-means clustering

- K-means is a partitional clustering algorithm
- Let the set of data points (or instances) D be

  $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$,

  where $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ir})$ is a vector in a real-valued space $X \subseteq R^r$, and r is the number of attributes (dimensions) in the data.
- The *k*-means algorithm partitions the given data into *k* clusters.
  - Each cluster has a cluster **center**, called **centroid**.
  - *k* is specified by the user

K-means Algorithm 1:

1. Decide on a value for *k*.

2. Initialize the *k* cluster centers (randomly, if necessary).

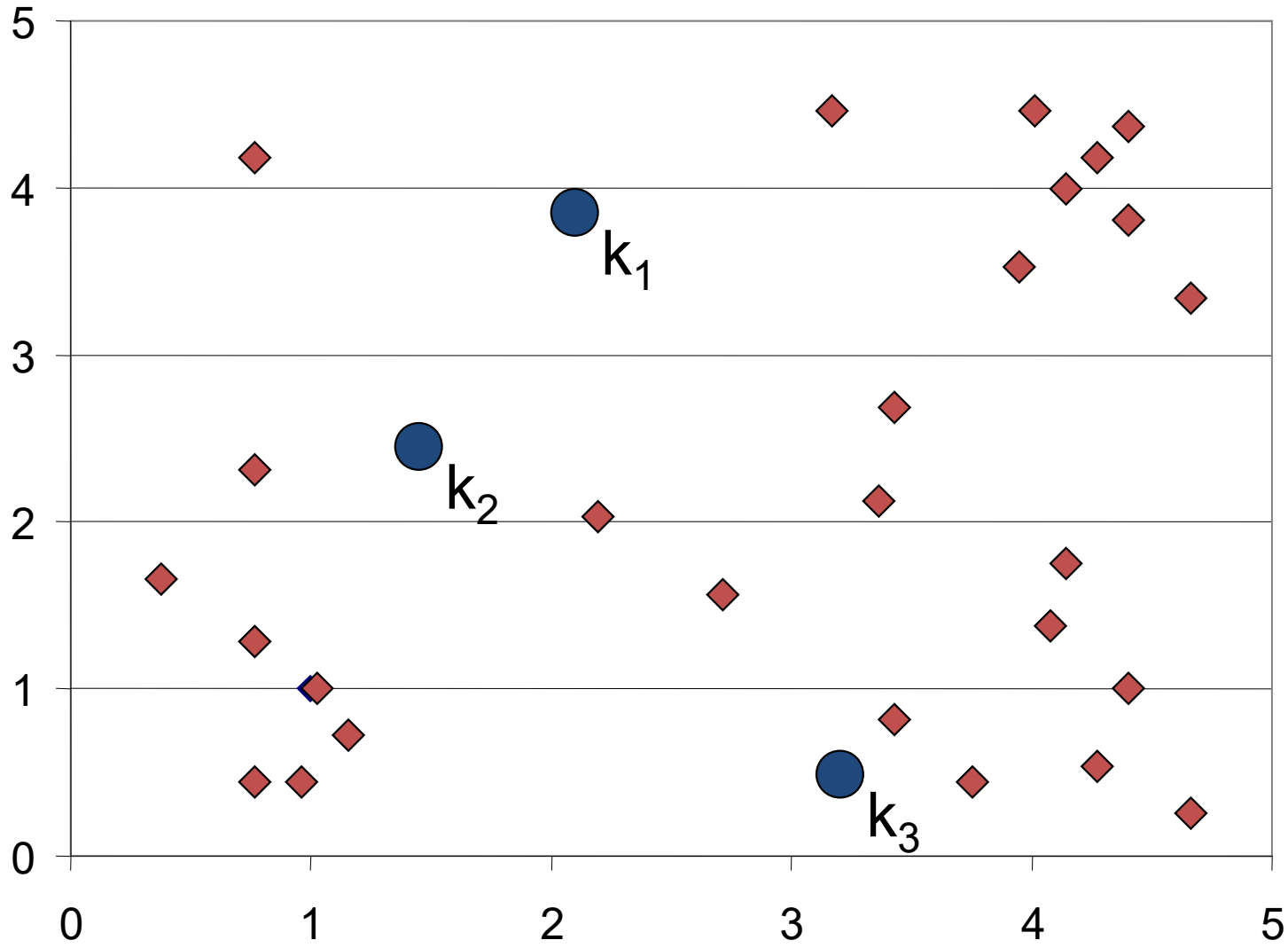
3. Decide the class memberships of the *N* objects by assigning them to the nearest cluster center.

4. Re-estimate the *k* cluster centers, by assuming the memberships found above are correct.

5. If none of the *N* objects changed membership in the last iteration, exit. Otherwise goto 3.

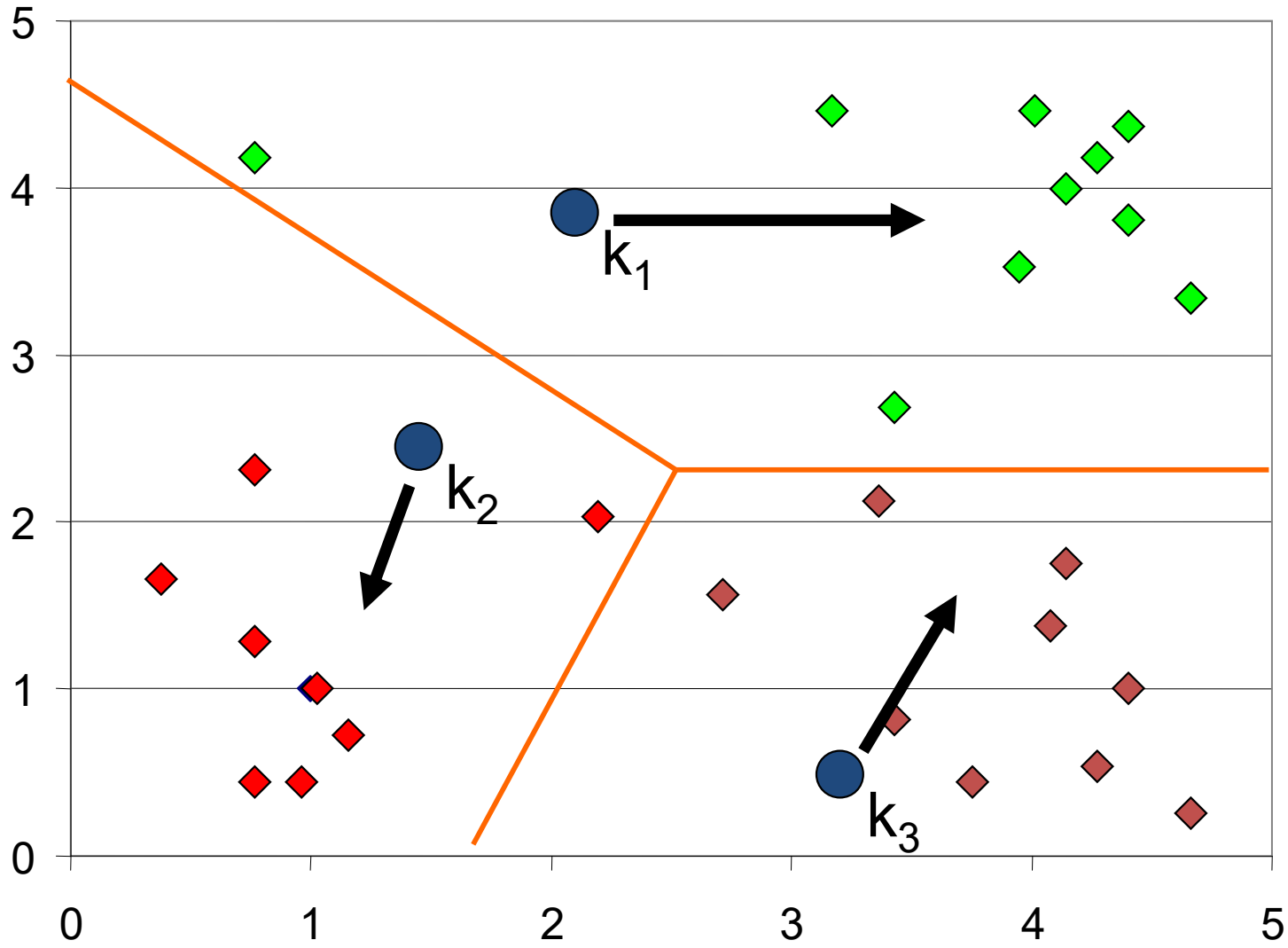
K-means Clustering: Step 1

Algorithm: k-means, Distance Metric: Euclidean Distance



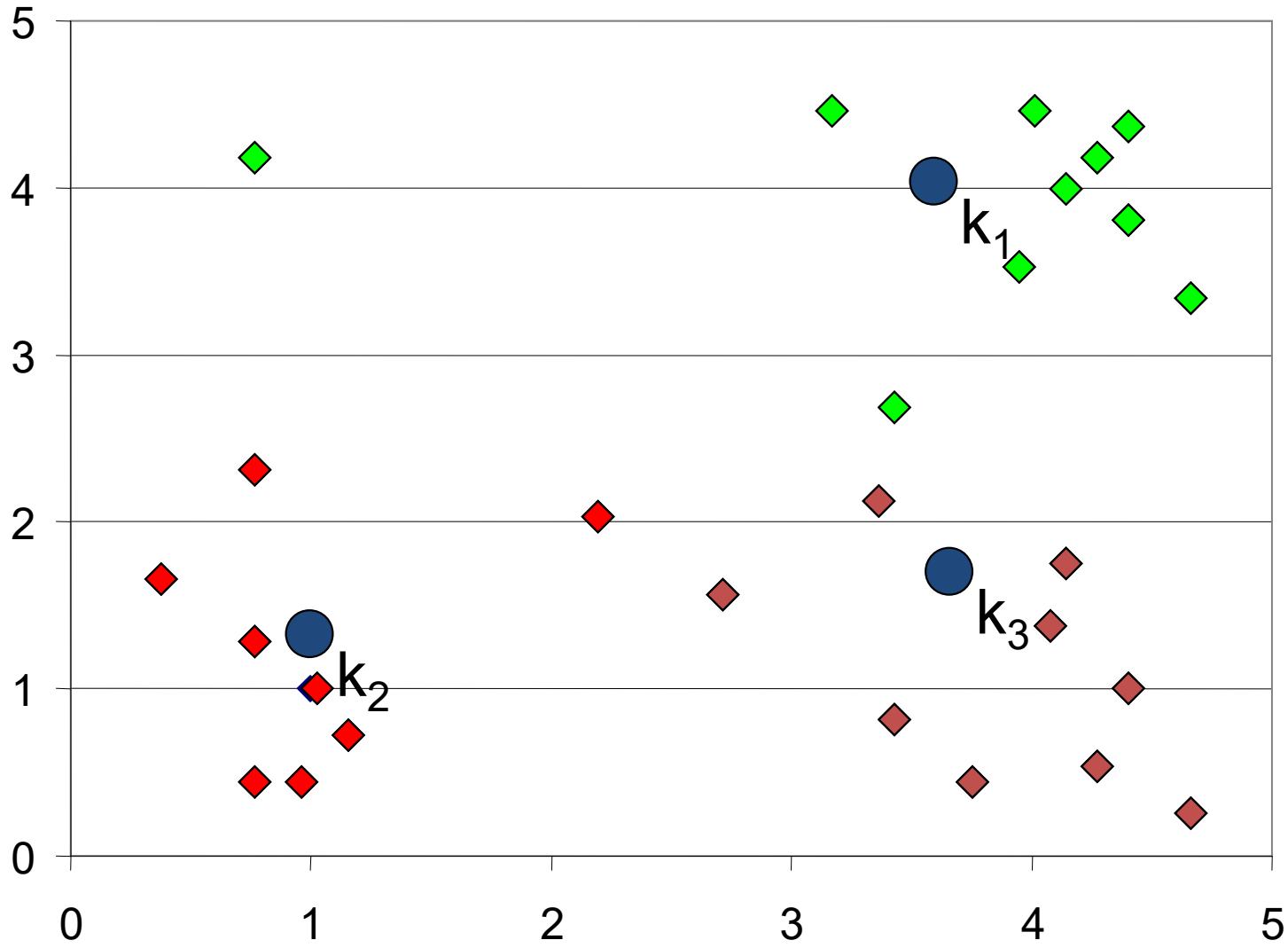
K-means Clustering: Step 2

Algorithm: k-means, Distance Metric: Euclidean Distance



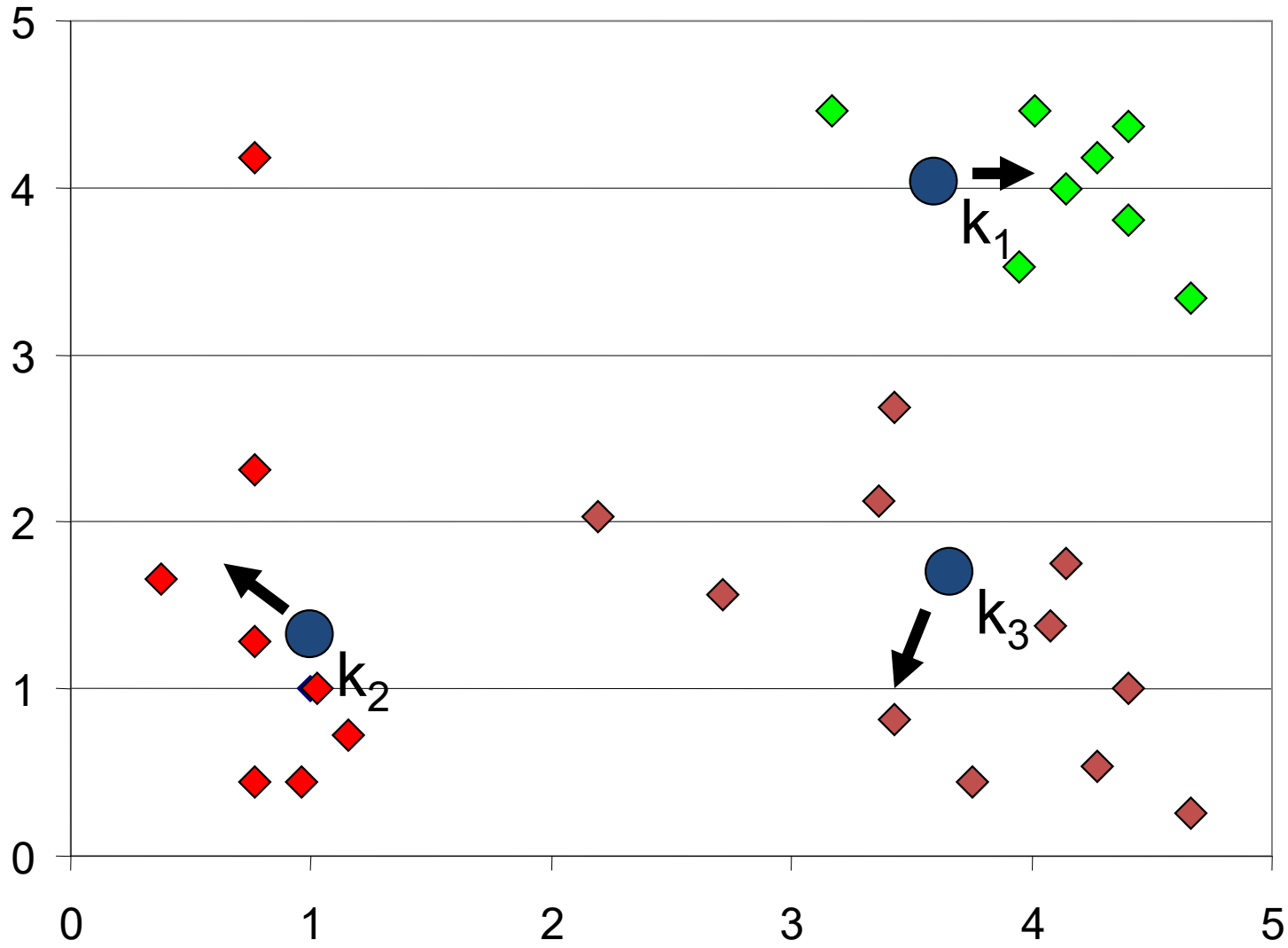
K-means Clustering: Step 3

Algorithm: k-means, Distance Metric: Euclidean Distance



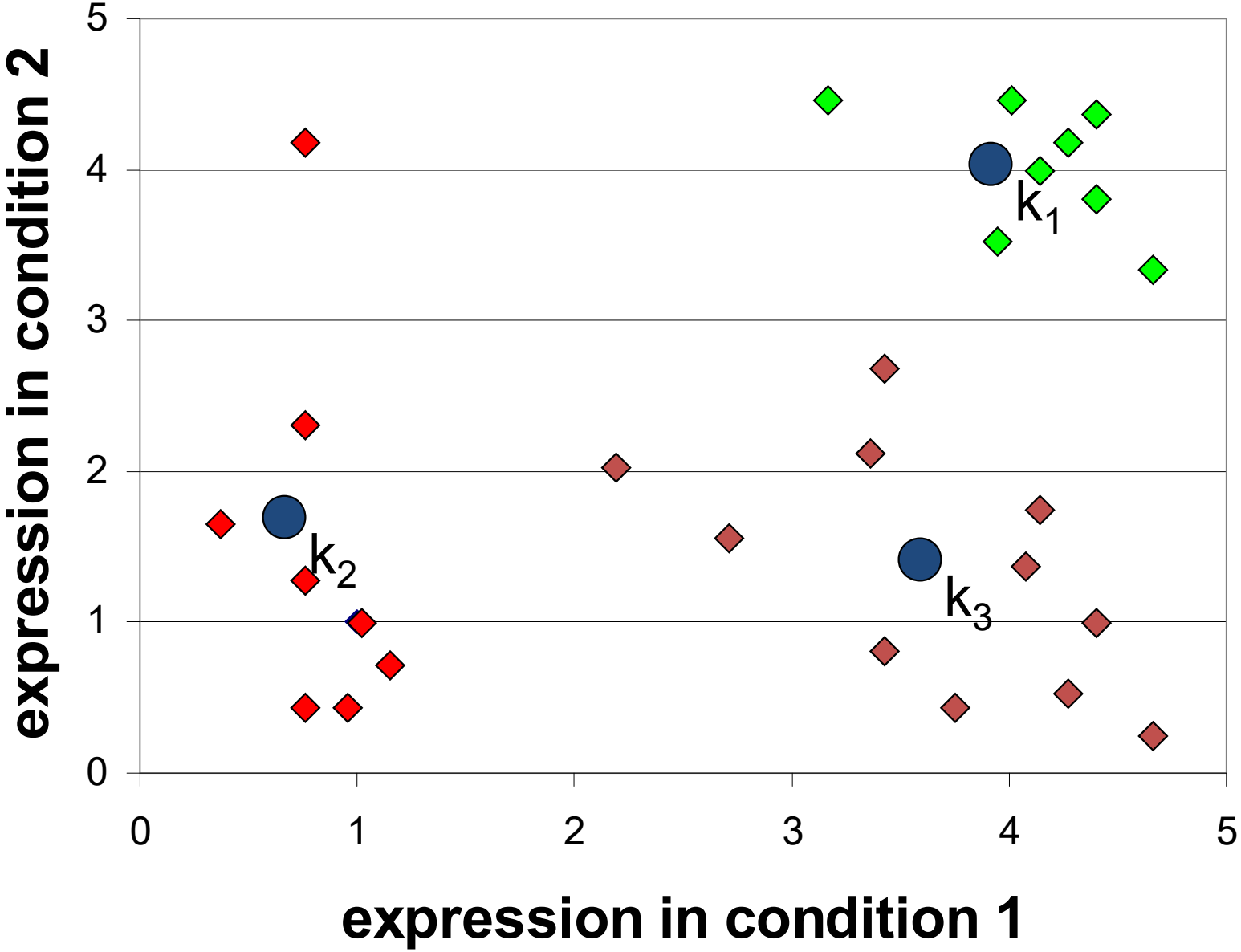
K-means Clustering: Step 4

Algorithm: k-means, Distance Metric: Euclidean Distance



K-means Clustering: Step 5

Algorithm: k-means, Distance Metric: Euclidean Distance



Mathematical Perspective

Given a set of observations (x1, x2, ..., xn), where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k (≤ n) sets S = {S1, S2, ..., Sk} so as to minimize the within-cluster sum of squares (WCSS) (i.e. variance). Formally, the objective is to find:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

where $\boldsymbol{\mu}_i$ is the mean (also called centroid) of points in S_i, i.e.

$$\boldsymbol{\mu}_i = \frac{1}{|S_i|} \sum_{\mathbf{x} \in S_i} \mathbf{x},$$

$|S_i|$ is the size of S_i, and $\|..\|$ is the usual L^2 norm

Clustering criterion ..

1. Similarity/distance function
2. Stopping criterion
3. Cluster Quality

1. Distance functions for numeric attributes

- Most commonly used functions are
  - Euclidean distance and
  - Manhattan (city block) distance
- We denote distance with: *dist*($\mathbf{x}_i$, $\mathbf{x}_j$), where $\mathbf{x}_i$ and $\mathbf{x}_j$ are data points (vectors)
- They are special cases of Minkowski distance. h is positive integer.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = ((x_{i1} - x_{j1})^h + (x_{i2} - x_{j2})^h + \ldots + (x_{ir} - x_{jr})^h)^{\frac{1}{h}}$$

Euclidean distance and Manhattan distance

- If h = 2, it is the Euclidean distance

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ir} - x_{jr})^2}$$

- If h = 1, it is the Manhattan distance

$$dist(\mathbf{x}_i, \mathbf{x}_j) = | x_{i1} - x_{j1} | + | x_{i2} - x_{j2} | + \ldots + | x_{ir} - x_{jr} |$$

- Weighted Euclidean distance

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{w_1(x_{i1} - x_{j1})^2 + w_2(x_{i2} - x_{j2})^2 + \ldots + w_r(x_{ir} - x_{jr})^2}$$

Squared distance and Chebychev distance

- Squared Euclidean distance: to place progressively greater weight on data points that are further apart.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = (x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ir} - x_{jr})^2$$

- Chebychev distance: one wants to define two data points as "different" if they are different on any one of the attributes.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \max(|x_{i1} - x_{j1}|, |x_{i2} - x_{j2}|, \ldots, |x_{ir} - x_{jr}|)$$

What properties should a distance measure have?

- $D(A,B) = D(B,A)$ *Symmetry*
- $D(A,A) = 0$ *Constancy of Self-Similarity*
- $D(A,B) = 0$ iif $A = B$ *Positivity (Separation)*
- $D(A,B) \leq D(A,C) + D(B,C)$ *Triangular Inequality*

Intuitions behind desirable distance measure properties

$D(A,B) = D(B,A)$

Otherwise you could claim "Alex looks like Bob, but Bob looks nothing like Alex."

$D(A,A) = 0$

Otherwise you could claim "Alex looks more like Bob, than Bob does."

Distance functions for binary and nominal attributes

- Binary attribute: has two values or states but no ordering relationships, e.g.,
  - Gender: male and female.
  - Weather: rain, sunny
- We use a confusion matrix to introduce the distance functions/measures.
- Let the *i*th and *j*th data points be $\mathbf{x}_i$ and $\mathbf{x}_j$ (vectors)

Confusion matrix

Data point i \ Data point j	1	0	
1	a	b	$a+b$
0	c	d	$c+d$
	$a+c$	$b+d$	$a+b+c+d$

(10)

a: the number of attributes with the value of 1 for both data points.

b: the number of attributes for which $x_{if} = 1$ and $x_{jf} = 0$, where x_{if} (x_{jf}) is the value of the fth attribute of the data point $\mathbf{x}_i$ ($\mathbf{x}_j$).

c: the number of attributes for which $x_{if} = 0$ and $x_{jf} = 1$.

d: the number of attributes with the value of 0 for both data points.

Symmetric binary attributes

- A binary attribute is **symmetric** if both of its states (0 and 1) have equal importance, and carry the same weights, e.g., male and female of the attribute Gender
- Distance function: Simple Matching Coefficient, proportion of mismatches of their values

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{b + c}{a + b + c + d}$$

Symmetric binary attributes: example

$\mathbf{x}_1$	1	1	1	0	1	0	0
$\mathbf{x}_2$	0	1	1	0	0	1	0

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{2+1}{2+2+1+2} = \frac{3}{7} = 0.429$$

Asymmetric binary attributes

- Asymmetric: if one of the states is more important or more valuable than the other.
  - By convention, state 1 represents the more important state, which is typically the rare or infrequent state.
  - Jaccard coefficient is a popular measure

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{b + c}{a + b + c}$$

  - We can have some variations, adding weights

Nominal attributes

- Nominal attributes: with more than two states or values.
  - the commonly used distance measure is also based on the simple matching method.
  - Given two data points $\mathbf{x}_i$ and $\mathbf{x}_j$, let the number of attributes be r, and the number of values that match in $\mathbf{x}_i$ and $\mathbf{x}_j$ be q.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{r - q}{r}$$

Normalization

Technique to force the attributes to have a common value range

what is the need ?

Consider the following pair of data points

$\mathbf{x}_i$: (0.1, 20) and $\mathbf{x}_j$: (0.9, 720).

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(0.9-0.1)^2 + (720-20)^2} = 700.000457,$$

The distance is almost completely dominated by (720-20) = 700.

Standardize attributes: to force the attributes to have a common value range

Interval-scaled attributes

- Their values are real numbers following a linear scale.
  - The difference in Age between 10 and 20 is the same as that between 40 and 50.
  - The key idea is that intervals keep the same importance through out the scale
- Two main approaches to standardize interval scaled attributes, **range** and **z-score**. *f* is an attribute

$$range(x_{if}) = \frac{x_{if} - \min(f)}{\max(f) - \min(f)},$$

Interval-scaled attributes (cont …)

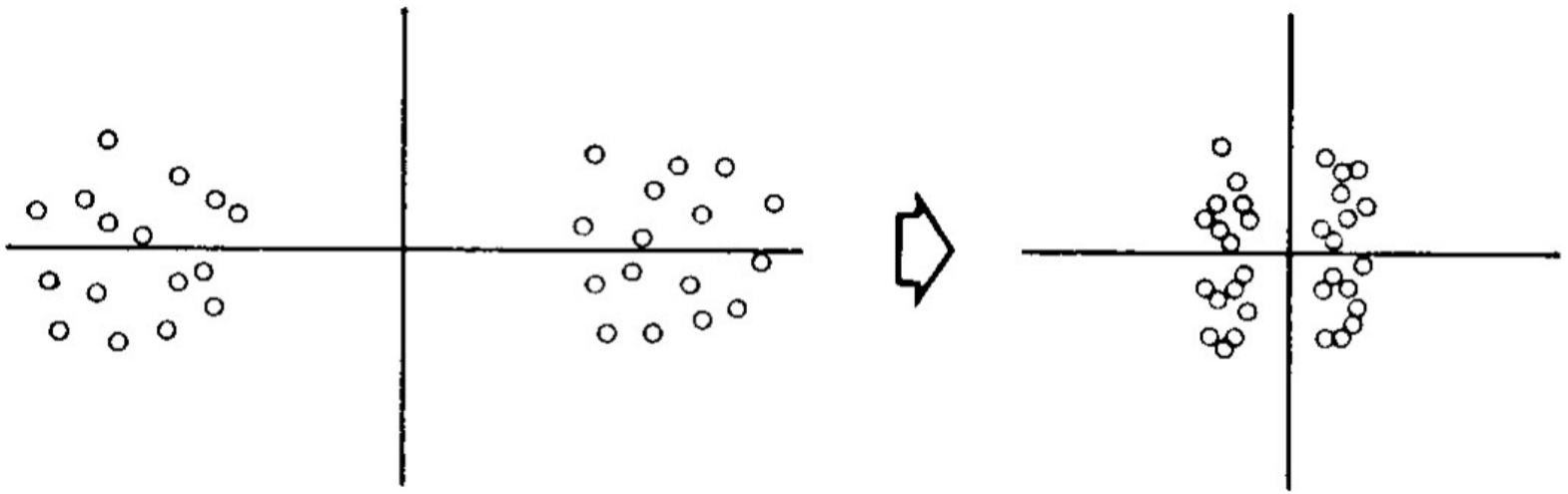
- Z-score: transforms the attribute values so that they have a mean of zero and a **mean absolute deviation** of 1. The mean absolute deviation of attribute *f*, denoted by s_f, is computed as follows

$$s_f = \frac{1}{n}\left(|x_{1f} - m_f| + |x_{2f} - m_f| + \ldots + |x_{nf} - m_f|\right),$$

$$m_f = \frac{1}{n}\left(x_{1f} + x_{2f} + \ldots + x_{nf}\right),$$

Z-score: $$z(x_{if}) = \frac{x_{if} - m_f}{s_f}.$$

Is normalization desirable?



(a) UNNORMALIZED

(b) NORMALIZED

Other distance/similarity measures

Distance between two instances x and x', where q >= 1 is a selectable parameter and d is the number of attributes (called the Minkowski Metric)

$$d(x,x') = \left(\sum_{j=1}^{d} \left| x_j - x'_j \right|^q\right)^{1/q}$$

Cosine of the angle between two vectors a similarity function:

$$s(x,x') = \frac{x^t x'}{\|x\| \|x'\|}$$

When features are binary this becomes the number of attributes shared by x and x' divided by the geometric mean of the number of attributes in x and the number in x'. A simplification of this is:

$$s(x,x') = \frac{x^t x'}{d}$$

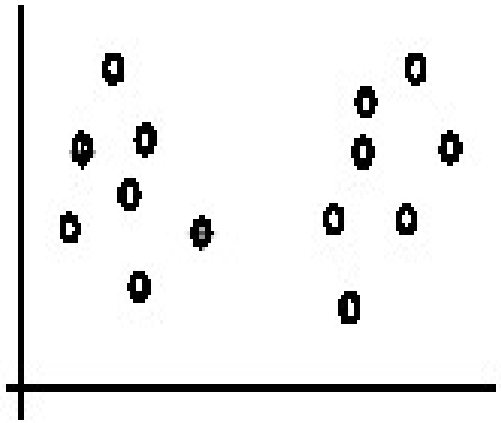
2. Stopping criteria

1. no (or minimum) re-assignments of data points to different clusters,
2. no (or minimum) change of centroids, or
3. minimum decrease in the **sum of squared error** (SSE),

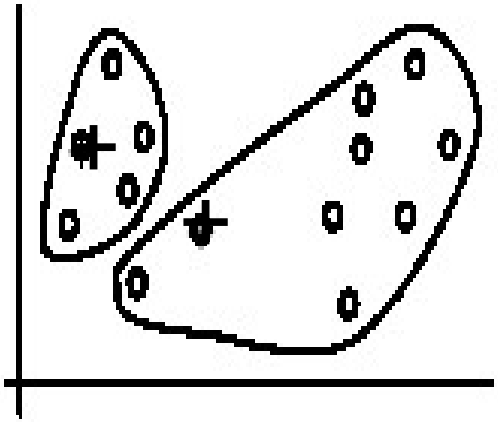
$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2$$

- C_i is the jth cluster, $\mathbf{m}_j$ is the centroid of cluster C_j (the mean vector of all the data points in C_j), and $dist(\mathbf{x}, \mathbf{m}_j)$ is the distance between data point $\mathbf{x}$ and centroid $\mathbf{m}_j$.

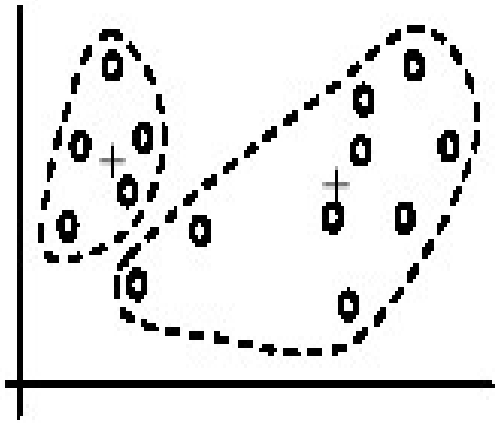
An example



(A). Random selection of k centers

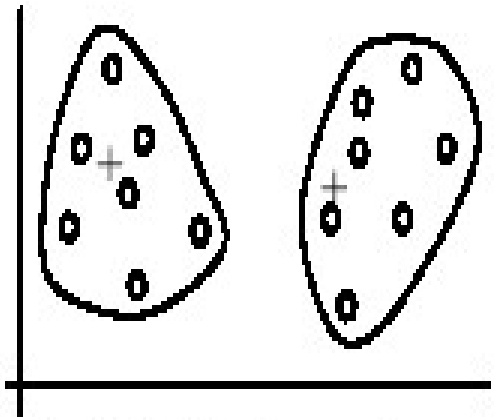


Iteration 1: (B). Cluster assignment

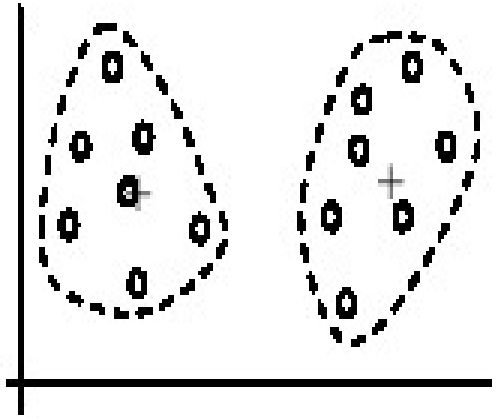


(C). Re-compute centroids

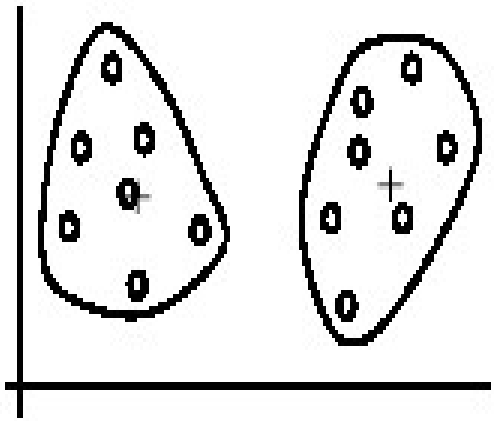
An example (cont …)



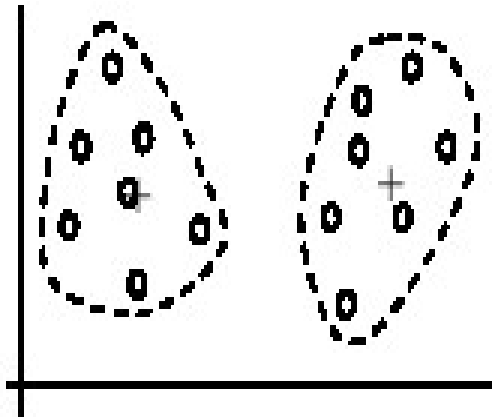
Iteration 2: (D). Cluster assignment



(E). Re-compute centroids



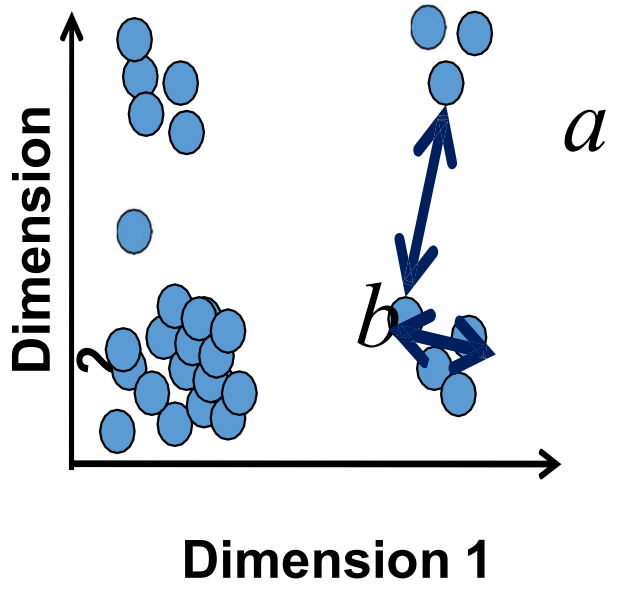
Iteration 3: (F). Cluster assignment



(G). Re-compute centroids

3. Cluster quality

- **Intra-cluster cohesion** (compactness):
  - Cohesion measures how near the data points in a cluster are to the cluster centroid.
  - Sum of squared error (SSE) is a commonly used measure.
- **Inter-cluster separation** (isolation):
  - Separation means that different cluster centroids should be far away from one another.



Cluster Evaluation: hard problem

- The quality of a clustering is very hard to evaluate because
  - We do not know the correct clusters
- Some methods are used:
  - User inspection
    - Study centroids, and spreads
    - Rules from a decision tree.
    - For text documents, one can read some documents in clusters.

Cluster evaluation: ground truth

- We use some labeled data (for classification)
- Assumption: Each class is a cluster.
- After clustering, a confusion matrix is constructed. From the matrix, we compute various measurements, entropy, purity, precision, recall and F-score.
  - Let the classes in the data D be $C = (c_1, c_2, \ldots, c_k)$. The clustering method produces k clusters, which divides D into k disjoint subsets, $D_1, D_2, \ldots, D_k$.

Evaluation measures: Entropy

Entropy: For each cluster, we can measure its entropy as follows:

$$entropy(D_i) = -\sum_{j=1}^{k} \Pr_i(c_j) \log_2 \Pr_i(c_j), \qquad (29)$$

where $\Pr_i(c_j)$ is the proportion of class c_j data points in cluster i or D_i. The total entropy of the whole clustering (which considers all clusters) is

$$entropy_{total}(D) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times entropy(D_i) \qquad (30)$$

Evaluation measures: purity

Purity: This again measures the extent that a cluster contains only one class of data. The purity of each cluster is computed with

$$purity(D_i) = \max_j(\Pr_i(c_j)) \tag{31}$$

The total purity of the whole clustering (considering all clusters) is

$$purity_{total}(D) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} \times purity(D_i) \tag{32}$$

Indirect evaluation

- In some applications, clustering is not the primary task, but used to help perform another task.
- We can use the performance on the primary task to compare clustering methods.
- For instance, in an application, the primary task is to provide recommendations on book purchasing to online shoppers.
  - If we can cluster books according to their features, we might be able to provide better recommendations.
  - We can evaluate different clustering algorithms based on how well they help with the recommendation task.
  - Here, we assume that the recommendation can be reliably evaluated.

Problems with Selecting Initial Points

- If there are K 'real' clusters then the chance of selecting one centroid from each cluster is small.
  - Chance is relatively small when K is large
  - If clusters are the same size, n, then

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select } K \text{ centroids}} = \frac{K!n^K}{(Kn)^K} = \frac{K!}{K^K}$$

  - For example, if K = 10, then probability = $10!/10^{10} = 0.00036$
  - Sometimes the initial centroids will readjust themselves in 'right' way, and sometimes they don't
  - Consider an example of five pairs of clusters

Comments on the *K-Means* Method

- Strength
  - *Relatively efficient*: *O*(*tkn*), where *n* is # instances, k is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
  - Often terminates at a *local optimum*. The *global optimum* may be found using techniques such as: *simulated annealing* or *genetic algorithms*
- Weakness
  - Applicable only when *mean* is defined; what about categorical data?
  - Need to specify c, the *number* of clusters, in advance
  - Unable to handle noisy data and *outliers*
  - Not suitable to discover clusters with *non-convex shapes*

Variations of the *K-Means* Method

- A few variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means
- Handling categorical data: *k-modes* (Huang'98)
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method

Agglomerative Hierarchical Clustering

Given a set of n instances to be clustered, and an nxn distance (or similarity) matrix, the basic process hierarchical clustering is:

- 1 Start by assigning each item to its own cluster, so that if you have N items, you now have N clusters, each containing just one item.
Let the distances (similarities) between the clusters equal the distances (similarities) between the items they contain.

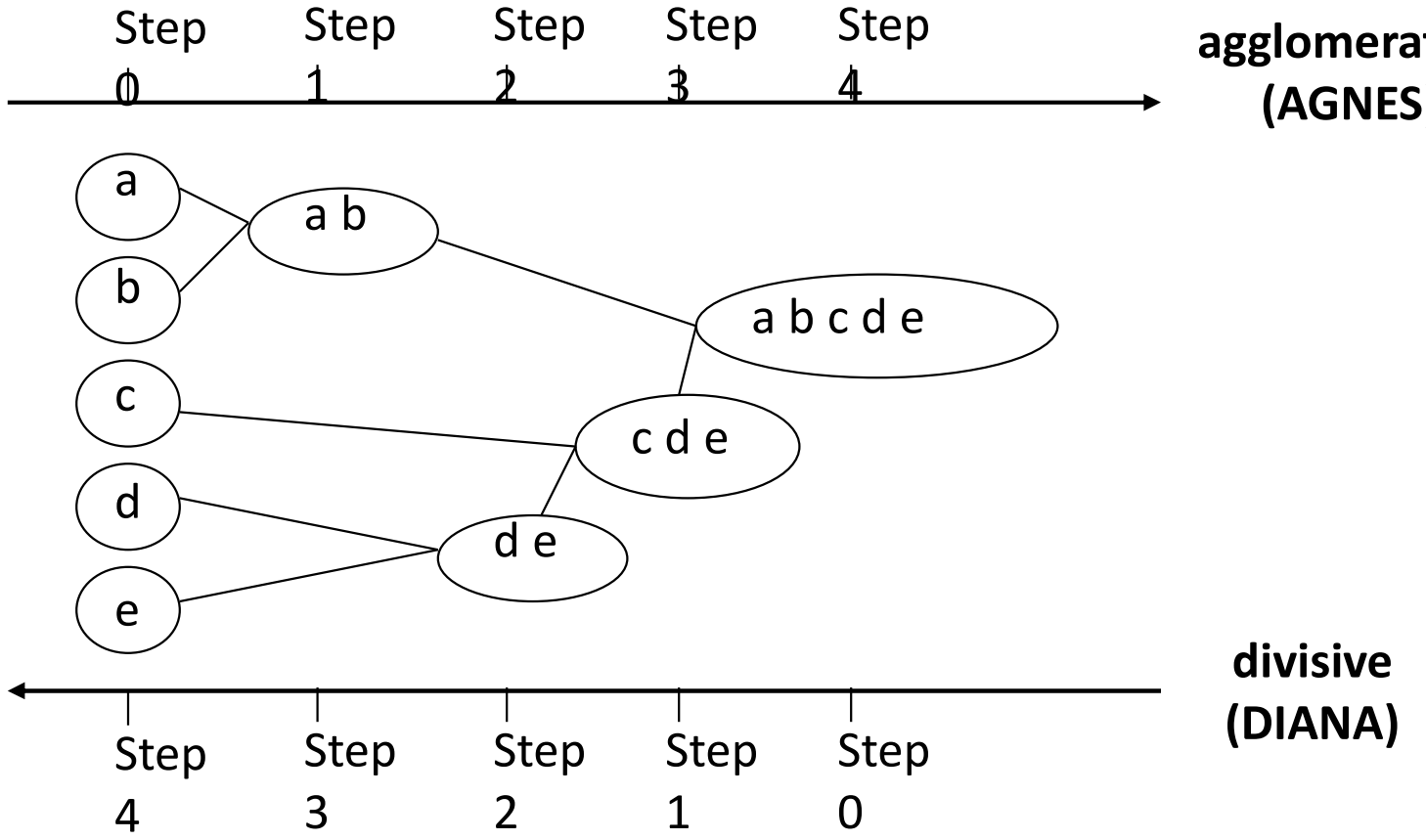
- 2 .Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one less cluster.

- 3. Compute distances (similarities) between the new clusters and each of the old clusters.

- 4. Repeat steps 2 and 3 until all items are clustered into a single cluster of size n.

Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition

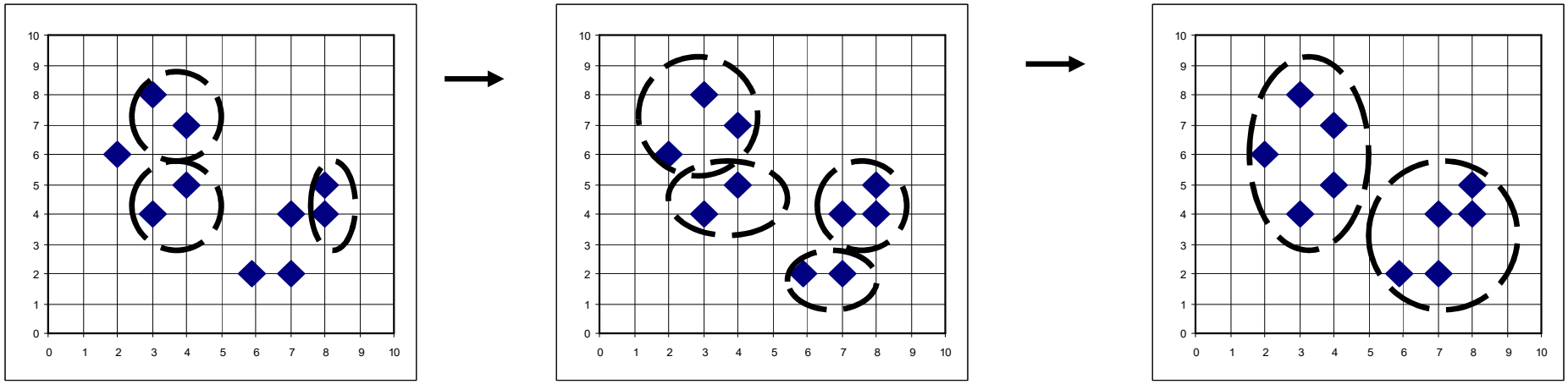


More on Hierarchical Clustering Methods

- Major weakness of agglomerative clustering methods
  - do not scale well: time complexity of at least $O(n^2)$, where n is the total number of instances
  - can never undo what was done previously
- Integration of hierarchical with distance-based clustering
  - BIRCH (1996): uses CF-tree and incrementally adjusts the quality of sub-clusters
  - CURE (1998): selects well-scattered points from the cluster and then shrinks them towards the center of the cluster by a specified fraction
  - CHAMELEON (1999): hierarchical clustering using dynamic modeling

AGNES (Agglomerative Nesting)

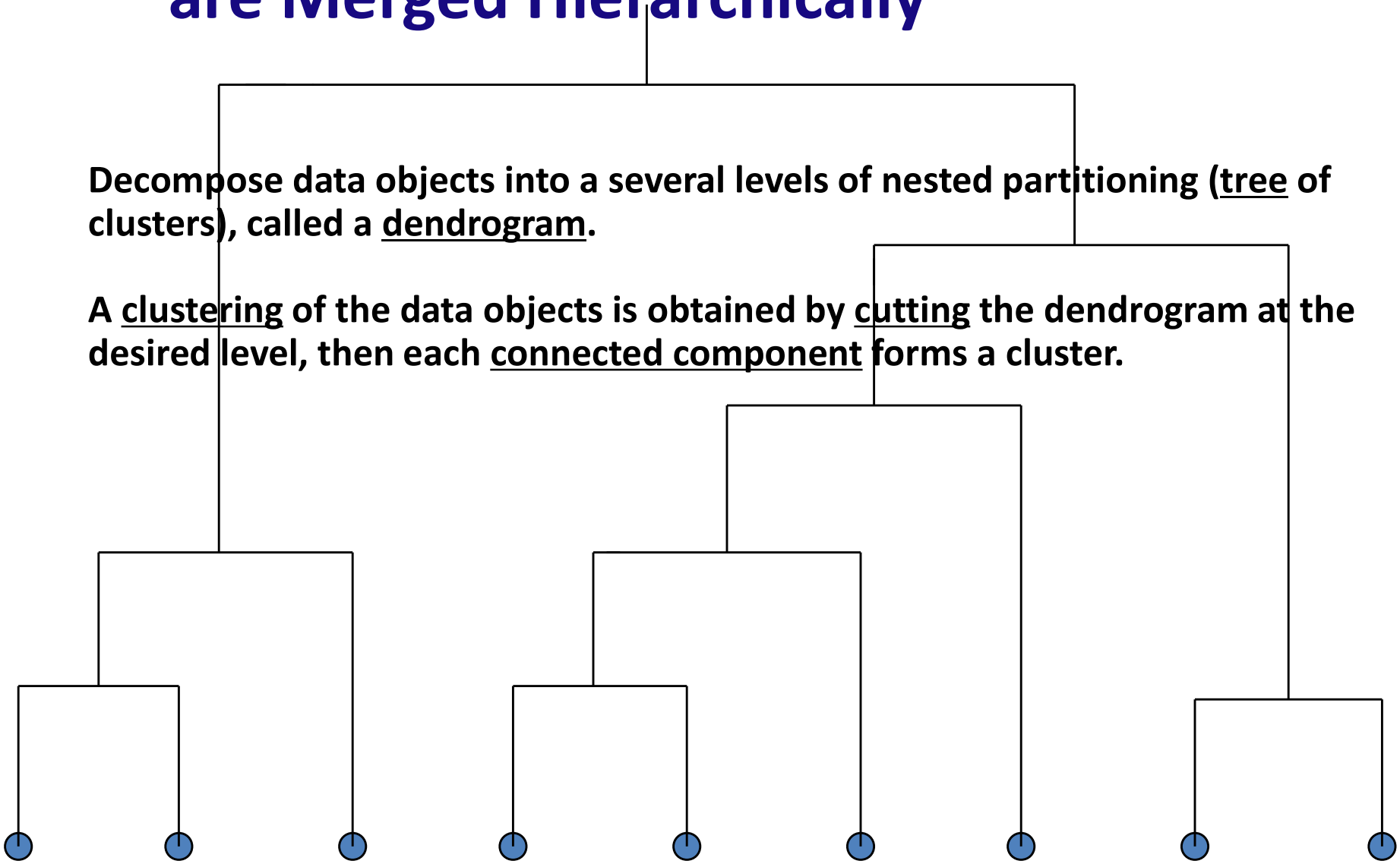
- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Use the Single-Link method and the dissimilarity matrix.
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



A *Dendrogram* Shows How the Clusters are Merged Hierarchically

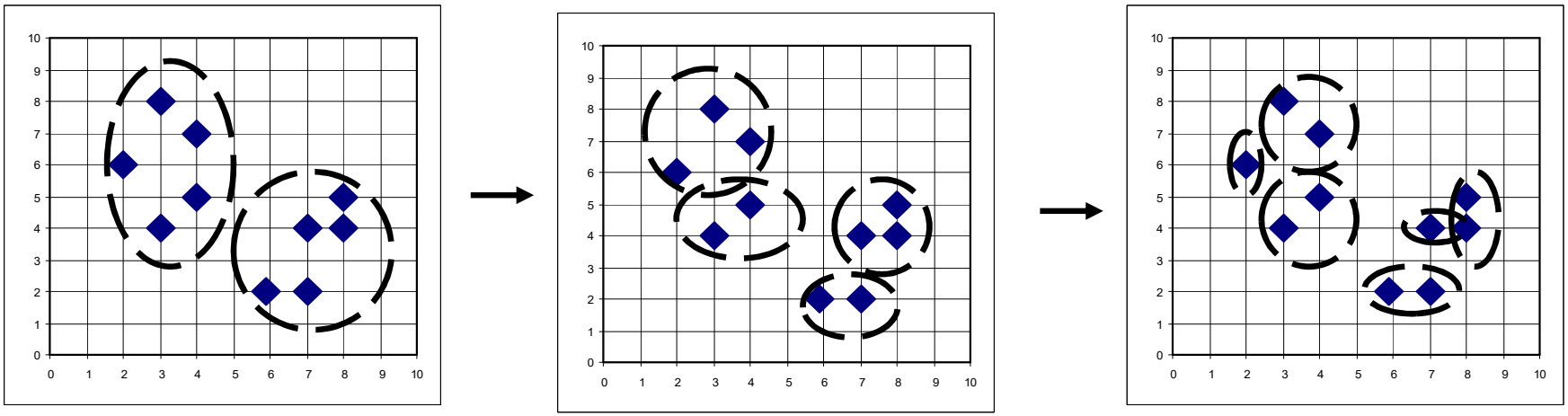
Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram.

A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.



DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own



Computing Inter-Cluster Distances

• **single-link clustering** (also called the connectedness or minimum method) : we consider the distance between one cluster and another cluster to be equal to the **shortest distance** from any member of one cluster to any member of the other cluster. If the data consist of similarities, we consider the similarity between one cluster and another cluster to be equal to the greatest similarity from any member of one cluster to any member of the other cluster.

• **complete-link clustering** (also called the diameter or maximum method): we consider the distance between one cluster and another cluster to be equal to the **longest distance** from any member of one cluster to any member of the other cluster.

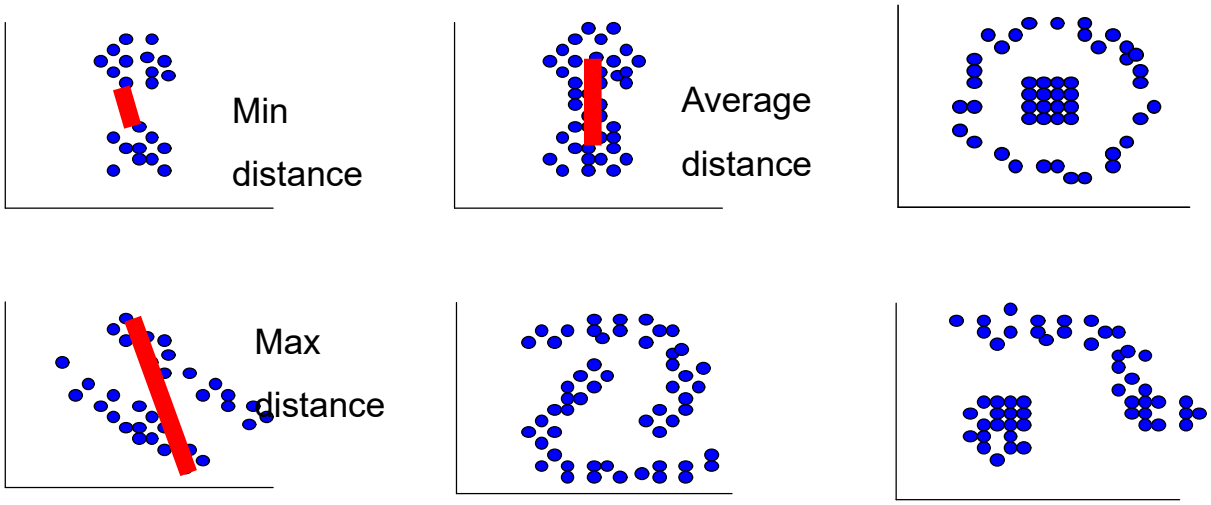
• **average-link clustering** : we consider the distance between one cluster and another cluster to be equal to the **average distance** from any member of one cluster to any member of the other cluster.

Distance Between Two Clusters

• single-link clustering (also called the connectedness or minimum method) : we consider the distance between one cluster and another cluster to be equal to the shortest distance from any member of one cluster to any member of the other cluster. If the data consist of similarities, we consider the similarity between one cluster and another cluster to be equal to the greatest similarity from any member of one cluster to any member of the other cluster.

• complete-link clustering (also called the diameter or maximum method): we consider the distance between one cluster and another cluster to be equal to the longest distance from any member of one cluster to any member of the other cluster.

• average-link clustering : we consider the distance between one cluster and another cluster to be equal to the average distance from any member of one cluster to any member of the other cluster.



- ❑Single-Link Method / Nearest Neighbor
- ❑Complete-Link / Furthest Neighbor
- ❑Their Centroids.
- ❑Average of all cross-cluster pairs.

Compare Dendrograms