Clustering Fundamentals

Q: What is clustering?

In cluster analysis, a group of objects is split up into a number of more or less homogeneous subgroups on the basis of an often subjectively chosen measure of similarity, such that the similarity between objects within a subgroup is larger than the similarity between objects belonging to different subgroups.

- Backer & Jain, 1981

Clustering vs. Classification

The fundamental difference between clustering and classification is that the clustering process does not use known class labels.

- Classification: The assignment of each sample to one of the pre-specified classes.
- Clustering: The division of (unlabeled) samples into clusters (groups).

Because the data are unlabeled, there is no absolute "correct" or "incorrect" way of clustering the data. As a result, "subjectivity" is somewhat unavoidable. Whether a particular clustering (partition) of the data is "good" or "bad" depends on the particular problem and need.

Clustering as Partition of Data

Let C_i be the i^{th} cluster of our data set X. it can be considered a set containing all the samples in it. Let there be a total of m clusters.

$$C_i \neq \phi, \ \forall i$$

$$C_1 \cup C_2 \cup ... \cup C_m = X$$

The following property makes this a hard (or crisp) partition (i.e., each $\boldsymbol{x}$ belongs to exactly one cluster):

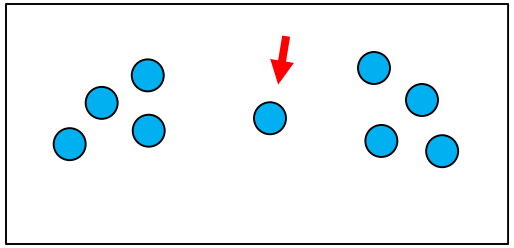
$$C_i \cap C_j = \phi, \ \forall i \neq j$$

The set of all the clusters for a dataset is usually called a ***clustering*** or a ***partition*** of the data:

$$\mathcal{R} = \{ C_1, C_2, ..., C_m \}$$

Fuzzy Partition of Data

How about if we have "ambiguous" samples?



We can also allow a sample to have partial memberships in multiple clusters. This leads to "**fuzzy partitions**" of the data. The conditions of partition of data become

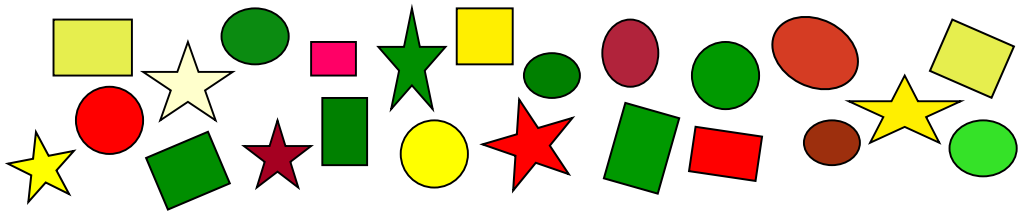
$$0 \le u_{ij} \le 1 \qquad \sum_i u_{ij} > 0, \ \forall j \qquad \sum_{j=1}^{m} u_{ij} = 1, \ \forall i$$

Here u_{ij} is called the membership of sample #i in cluster #j.

The last condition specifies a **probabilistic partition**. It may or may not be required depending on the clustering algorithm.

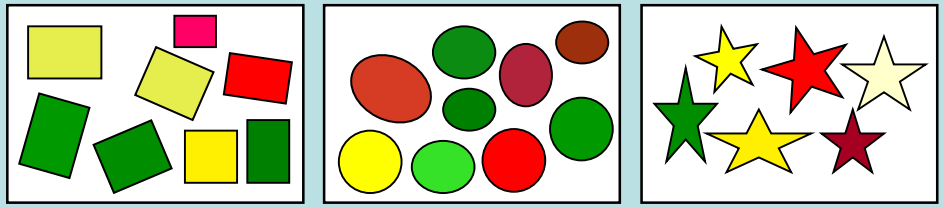
Clustering Criterion

Example: How do we cluster these shapes?

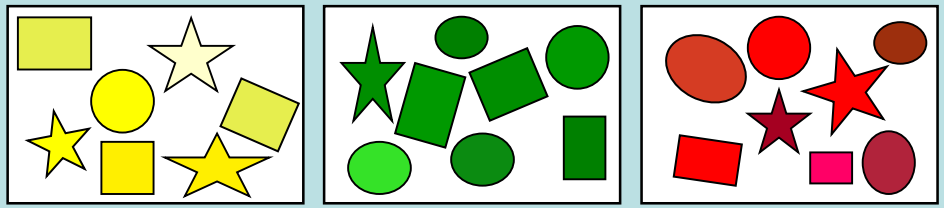


Clustering criterion is what we want to base the clustering on. Intuitively, it can be "color", "shape", or many other possible properties of the data.

By shape:



By color:



In practice, the clustering criterion can be formulated in many different ways. For example, we can try to minimize a cost function that corresponds to the criterion. The exact choice depends on the algorithm and the type of clusters desired.

Proximity Measure

- Since our goal is to group "similar" patterns into clusters, we need a measure of how similar/dissimilar two samples are.
- The choice of proximity measure directly affects the "shape" of clusters in the feature space.
- Whether to use a similarity or dissimilarity measure depends on the algorithm used.
- A dissimilarity measure is a “distance measure” if it satisfies triangular inequality.
- Similarity measures are often scaled to 0~1.
- We can always derive a similarity measure from a dissimilarity measure, such as

$$s(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-d^2(\boldsymbol{x}, \boldsymbol{y}) / d_0^2\right)$$

- We can always derive a dissimilarity measure from a similarity measure, such as

$$d(\boldsymbol{x}, \boldsymbol{y}) = 1 - s(\boldsymbol{x}, \boldsymbol{y})$$

Point-Set Proximity

- In some clustering algorithms, it is necessary to have a proximity function between a sample $\boldsymbol{x}$ and a cluster C. This allows procedures like finding the best cluster to assign $\boldsymbol{x}$ to.
- Let $\mathcal{P}(\boldsymbol{x},\boldsymbol{y})$ be any point-point proximity measure. Common corresponding point-set proximity functions are

$$\mathcal{P}_{max}(\boldsymbol{x}, C) = \max_{\boldsymbol{y}\in C} \mathcal{P}(\boldsymbol{x}, \boldsymbol{y})$$

$$\mathcal{P}_{min}(\boldsymbol{x}, C) = \min_{\boldsymbol{y}\in C} \mathcal{P}(\boldsymbol{x}, \boldsymbol{y})$$

$$\mathcal{P}_{avg}(\boldsymbol{x}, C) = \frac{1}{n_C} \sum_{\boldsymbol{y}\in C} \mathcal{P}(\boldsymbol{x}, \boldsymbol{y})$$

- When the cluster has representative $\boldsymbol{m}$:

$$\mathcal{P}(\boldsymbol{x}, C) = \mathcal{P}(\boldsymbol{x}, \boldsymbol{m})$$

Set-Set Proximity

- Cluster-cluster proximity is generally defined using the proximity between their members. They are used in clustering algorithms that require the merging or splitting of clusters during the process. Some common approaches:

$$\mathcal{P}_{max}(C_1, C_2) = \max_{\boldsymbol{x} \in C_1, \boldsymbol{y} \in C_2} \mathcal{P}(\boldsymbol{x}, \boldsymbol{y})$$

$$\mathcal{P}_{min}(C_1, C_2) = \min_{\boldsymbol{x} \in C_1, \boldsymbol{y} \in C_2} \mathcal{P}(\boldsymbol{x}, \boldsymbol{y})$$

$$\mathcal{P}_{avg}(C_1, C_2) = \frac{1}{n_{C_1} n_{C_2}} \sum_{\boldsymbol{x} \in C_1} \sum_{\boldsymbol{y} \in C_2} \mathcal{P}(\boldsymbol{x}, \boldsymbol{y})$$

- For clusters with representatives:

$$\mathcal{P}_{mean}(C_1, C_2) = \mathcal{P}(\boldsymbol{m}_{C_1}, \boldsymbol{m}_{C_2})$$

Clustering Relational Data

- It is not necessary to represent the samples with feature vectors.
- The clustering process uses only the "relations" (similarities / dissimilarities) among the samples.
- Particularly useful in many applications where it is difficult represent samples with feature vectors. Examples:
  - Biology
  - Document, music, etc
  - Network structure (e.g., social network)
  - …

First Algorithm: Sequential Clustering

This is a very simple but still practical procedure, particularly for large datasets that do not fit into the memory and streaming.

Basic ideas:

- Samples to be clustered are presented to the algorithm one by one.
- The decision to place a new sample $\boldsymbol{x}$ in a new cluster or one of the existing clusters is made at the time when $\boldsymbol{x}$ is presented.
- Hyperparameters:
  - A threshold of similarity/dissimilarity to be used in the decision.
  - The maximum number of generated clusters (optional).

Basic Sequential Clustering

Assume that each cluster has a representative (prototype).

$m \leftarrow 1$
$C_m \leftarrow \{\boldsymbol{x}_1\}$ first cluster
For $i = 2$ to N
  $k \leftarrow \text{argmin } d(\boldsymbol{x}_i, C_k)$
  If $(d(\boldsymbol{x}_i, C_k) > \Theta)$ AND $(m < q)$
    $m \leftarrow m + 1$
    $C_m \leftarrow \{\boldsymbol{x}_i\}$ make a new cluster
  Else
    $C_k \leftarrow C_k \cup \{\boldsymbol{x}_i\}$ add to an existing cluster
    Update cluster representatives if necessary
  End
End

Θ=3 and q=6; data: 1 2 3 5 7 8

{1}
{1,2} mean=1.5
{1,2,3} mean=2
{1,2,3,5} mean=2.7

{1,2,3,5}:2.7, {7}:7
{1,2,3,5}:2.7, {7,8}:7.5

Θ=3 and q=6; data: 3 5 7 1 2 8

{3}:3
{3,5}:4
{3,5,7}:5
{3,5,7}:5, {1}:1
{3,5,7}:5, {1,2}:1.5
{3,5,7,8}:5.7, {1,2}:1.5

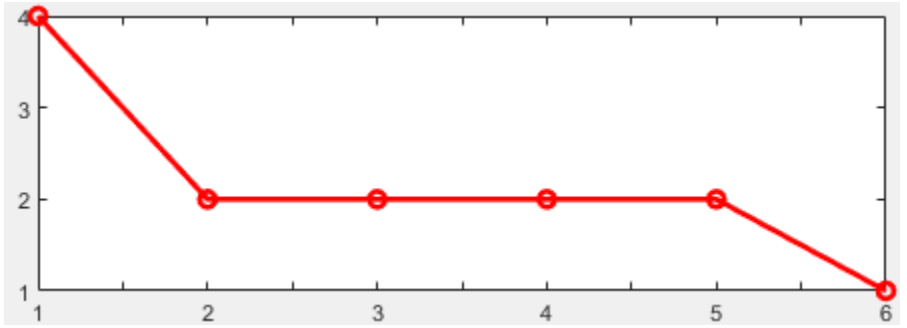
Selection of Clustering

- We find that different values of threshold Θ result in different clusterings (partitions). How do we choose a "good" one?
- Idea: Find a clustering that is most *insensitive* to the threshold.
- Method: Generate clusterings with different thresholds, and plot the resulting numbers of clusters (m) as a function of Θ . Flat regions in the plot indicate the existence of a stable clustering.

Example: Data: 1 2 3 6 7 8

Use Θ=1, 2, …, 7, and q=6

Present the data in the order: 3 2 8 7 1 6

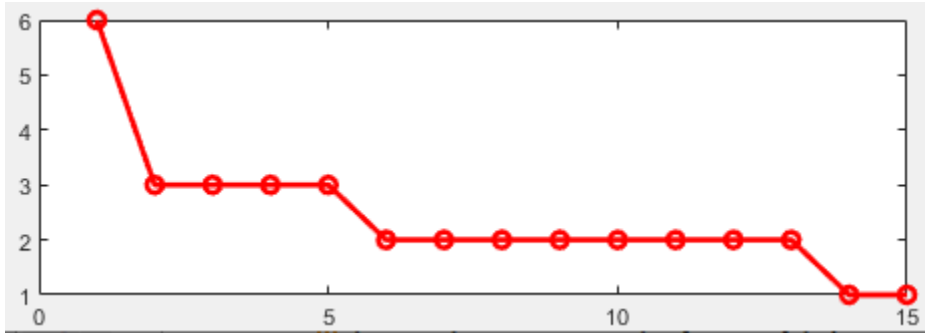


Example: Data: 1 2 3 7 8 9 15 16 17 18

Use Θ=1, 2, …, 15, and q=10

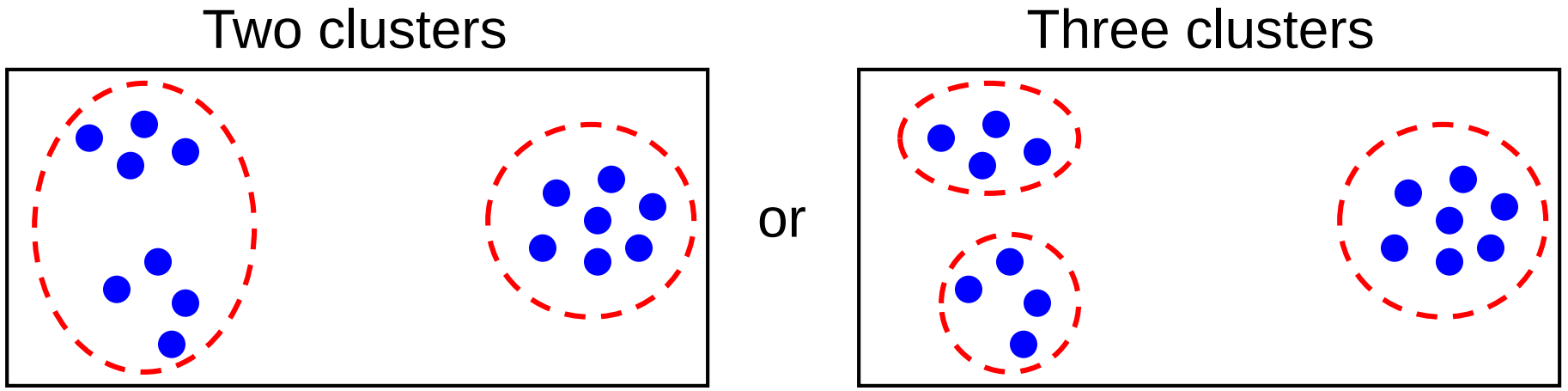
Present the data in the order:

3 9 2 18 17 8 7 16 1 15



Selection of Clustering

Sometimes there are more than one flat regions, indicating more than one stable ways to partition the data. The following is a possible scenario:



Both clusterings are considered reasonable.

Cluster Validity

There is no "correct answer" regarding the results of clustering algorithms. How can we identify the good ones from the possible clusterings?

Cluster Validity: A measure of whether a clustering is a reasonable representation of the actual "cluster structure" of the data.

Note: There are many cluster validity measures, and they are only tools that help us make the choice, not absolute standards.

Many cluster validity measures use ideas similar to class separability measures.

Internal and External Validity

- **Internal cluster validity measures:**
  - The validity measure is computed based on the clustering results only; no extra information used.
  - The objective is to check whether the resulting clusters are actually reasonable clusters.
  - Difficult to compare different validity measures.
  - Used to choose among different partitions of a dataset, or the hyper-parameters of the clustering algorithms used.

Internal and External Validity

- **External cluster validity measures:**
  - Use some external information (e.g., class labels) as a reference, so that we can check how well the clustering results coincide with the existing structure (e.g., classes) in the dataset.
  - Often used to compare the performance of different clustering algorithms. Otherwise it is difficult to compare them objectively.
  - It is assumed that the reference information, such as the classes, actually exhibit cluster-like structures, although this may or may out be the case.
  - Simulated datasets are often employed for this purpose.

Selecting Numbers of Clusters

A common and important use of internal cluster validity is for the determination of the number of clusters.

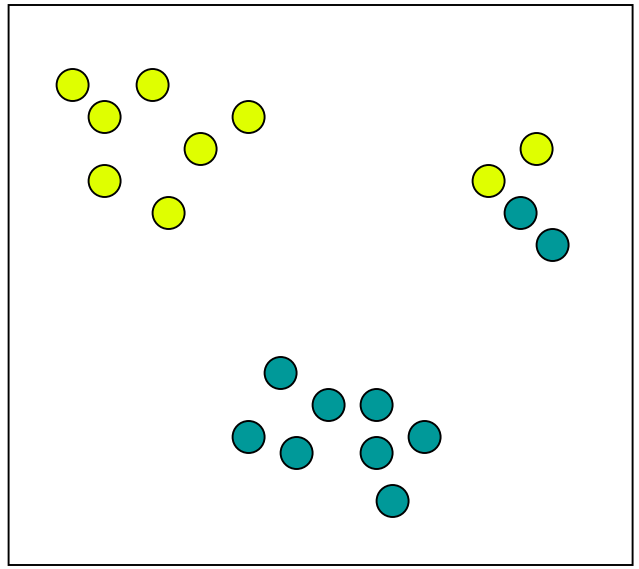
- Clustering algorithms that do not require a pre-specified number of clusters:
  - If we obtain a particular number of clusters with a wide range of parameters, then that number likely corresponds to the actual cluster structure of the data.
- Clustering algorithms that require a pre-specified number of clusters:
  - Run the algorithm with different numbers of clusters, and choose the number of clusters that give the best result of a certain cluster validity measure.

Internal Cluster Validity - Example

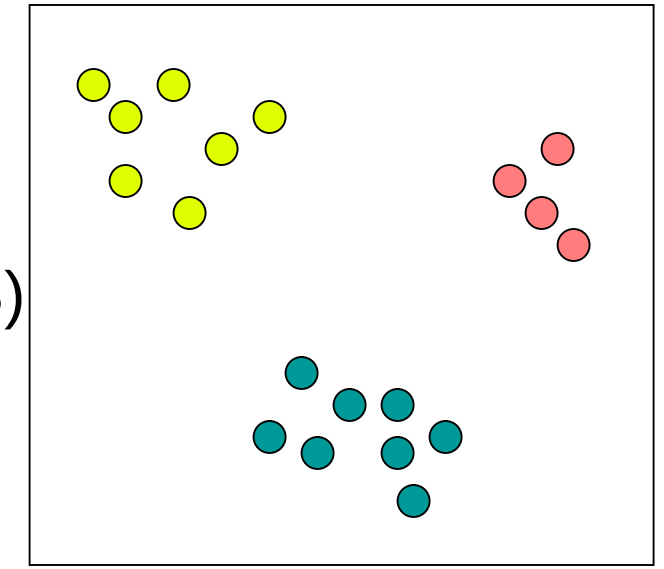
Two clusterings of the same data:

(B) should have better cluster validity than (A).

(A)



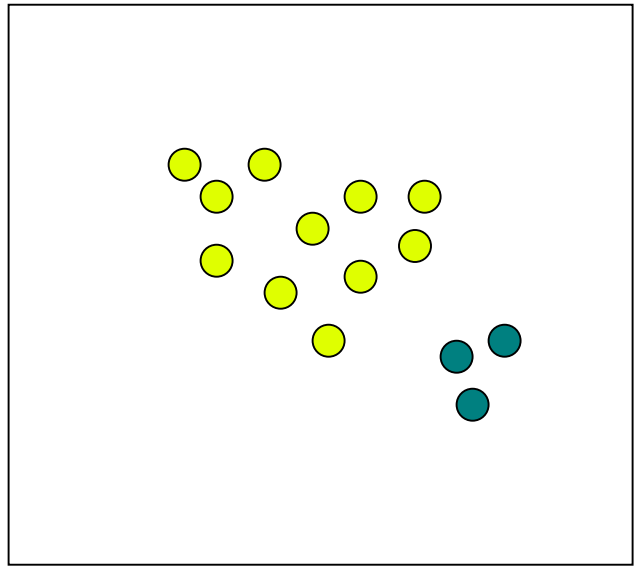
(B)



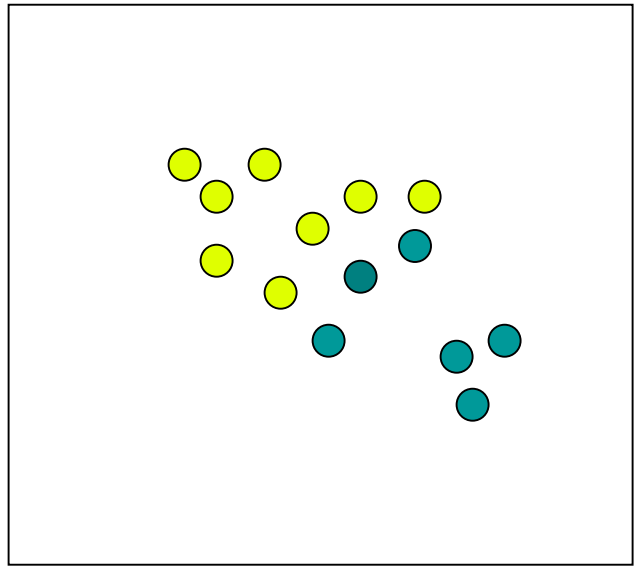
Two clusterings of the same data:

(A) should have better cluster validity than (B).

(A)



(B)



Internal Cluster Validity

The most common approaches for designing an internal cluster validity measure are based on the following idea:

To maximize

$$\frac{\text{Separation (distances) between clusters}}{\text{Scatter (sizes) of clusters}}$$

Or to minimize its inverse in some form.

There are dozens of cluster validity indices based on this concept. We will list only a few representative ones.

Note that some of them utilize the cluster prototypes / centroids / representatives, and others are computed directly from cluster memberships / assignments.

Dunn's Index

This cluster validity measure is specifically designed for hard (crisp) clusterings.

$$D_m = \frac{\min_{i \neq j} d(C_i, C_j)}{\max_k \left[diam(C_k) \right]}$$

The larger, the better.

Dissimilarity between clusters: $d(C_i, C_j) = \min_{\boldsymbol{x} \in C_i, \boldsymbol{y} \in C_j} d(\boldsymbol{x}, \boldsymbol{y})$

Cluster diameter: $diam(C_k) = \max_{\boldsymbol{x}, \boldsymbol{y} \in C_k} d(\boldsymbol{x}, \boldsymbol{y})$

Example: 1-D Data: 1 2 3 6 7 8

{1,2} {3,6} {7,8} ➔ 1/3

{1,2,3} {6,7,8} ➔ 3/2

Davies-Bouldin (DB) Index

This is designed for clusters with point prototypes:

$$DB_m = \frac{1}{m}\sum_{i=1}^{m}\max_{j\neq i}\left[\frac{s_i + s_j}{d(\boldsymbol{v}_i, \boldsymbol{v}_j)}\right]$$

The smaller, the better.

where $$s_j = \frac{1}{n_j}\sum_{\boldsymbol{x}\in C_j} d(\boldsymbol{x}, \boldsymbol{v}_j)$$ (for crisp clusters)

$$s_j = \frac{\sum_{i=1}^{N} u_{ij}^{q}\, d(\boldsymbol{x}_i, \boldsymbol{v}_j)}{\sum_{i=1}^{N} u_{ij}^{q}}$$ (for fuzzy clusters)

Xi-Beni (XB) Index

The ratio between the mean squared distances of the points to their assigned clusters and the minimum squared distances between cluster pairs. (The smaller, the better.)

$$XB = \frac{\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{c} u_{ij}^2 d^2(\boldsymbol{x}_i, \boldsymbol{v}_j)}{\min_{j \neq i} d^2(\boldsymbol{v}_i, \boldsymbol{v}_j)}$$

This form is naturally suitable for fuzzy clusters. It can be applied to crisp clusters by forcing u_{ij} to be 0 or 1.

Silhouette Index

The idea here is to maximize the difference of the data points' distances to points in other clusters relative to the distances to points in the same cluster. (The larger, the better.)

$$SI = \frac{1}{N} \sum_{j=1}^{c} \sum_{\boldsymbol{x} \in C_j} \frac{b(\boldsymbol{x}, C_j) - a(\boldsymbol{x}, C_j)}{\max[b(\boldsymbol{x}, C_j), a(\boldsymbol{x}, C_j)]}$$

where

$$a(\boldsymbol{x}_i, C_j) = \frac{1}{|C_j| - 1} \sum_{\boldsymbol{x}_k \in C_j} d^2(\boldsymbol{x}_i, \boldsymbol{x}_k)$$

$$b(\boldsymbol{x}_i, C_j) = \min_{C_r \neq C_j} \frac{1}{|C_r|} \sum_{\boldsymbol{x}_k \in C_r} d^2(\boldsymbol{x}_i, \boldsymbol{x}_k)$$

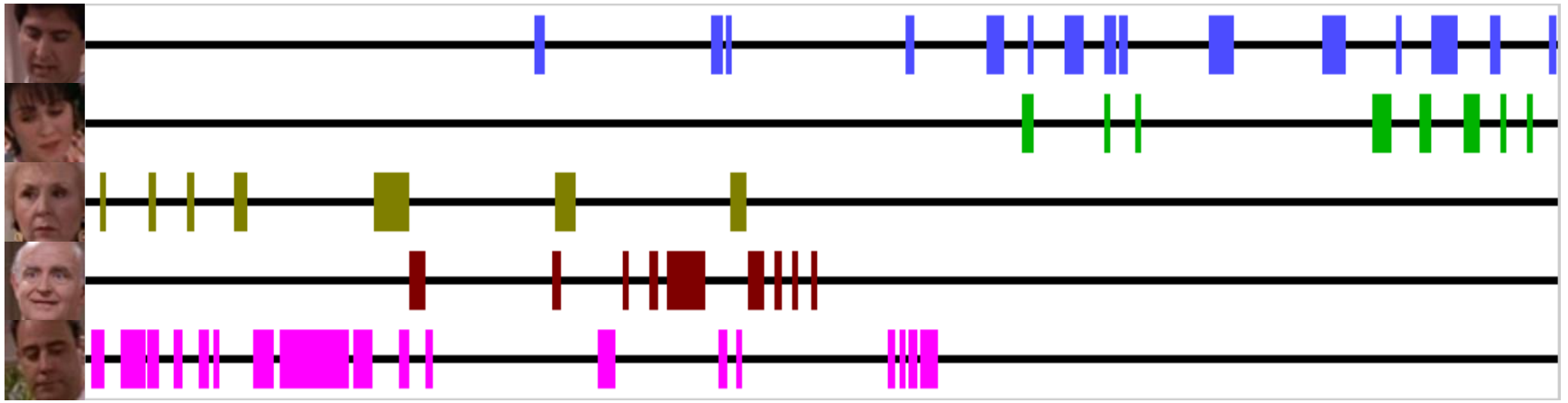
Objective Evaluation of Clustering

- In some problems, although we use clustering to process the data, there exist actual "correct" partitions of the data.
- The objective correctness of the partition can be evaluated by comparison with the correct partition.
- Methods of computing the similarity between two different partitions of the same data (e.g., the partition by the clustering algorithm, and the correct partition) are required.
- Since the correct partition is known, the purpose is not to find a clustering of the data at hand, but to evaluate the suitability of the clustering method for similar data.

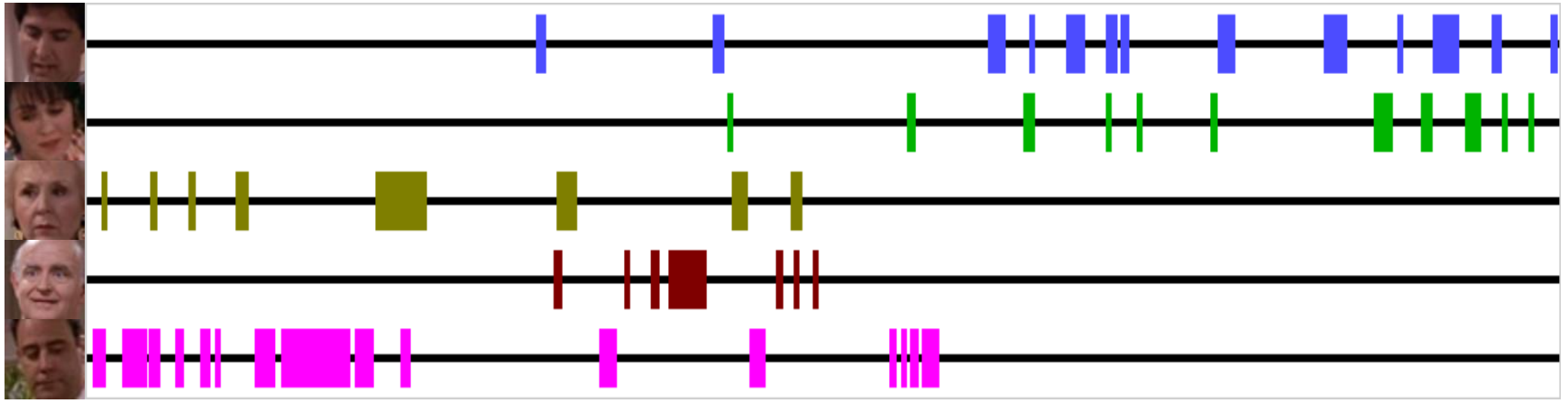
Objective Evaluation of Clustering

- Example application: Finding the main characters in a movie.

Ground Truth (200s)



Clustering Result (200s)



Objective Evaluation of Clustering

- **RAND Index** (RI) :
  - This is evaluated in a way similar to a binary classification problem, where each item to be classified is a pair of data points in the original data set.

    ➔ There are C_2^n items (pairs) to be classified, n being the number of data points.

$$RI = (TP + TN) / C_2^n$$

  - Positive: The two data points of a pair belong to the same subset (cluster).
  - Negative: The two data points of a pair belong to different subsets (clusters).
  - Range of value: 0~1.

Objective Evaluation of Clustering

- **Adjusted RAND Index** (ARI) :
  - The expected value of Rand Index for a random partition is not zero, and it depends on the number of clusters.
  - ARI is an corrected-for-chance version of RI.
  - Range of value: –1 to 1
  - Is zero when one partition
    - Contains all singleton clusters
    - Has only a single cluster
    - Is formed randomly (expected value)

Objective Evaluation of Clustering

- **Adjusted RAND Index** (ARI) :
  - Computed using the contingency matrix between two partitions X and Y:

$$n_{ij} = \left| X_i \cap Y_j \right|$$

$X \backslash Y$	Y_1	Y_2	$\cdots$	Y_s	sums
X_1	n_{11}	n_{12}	$\cdots$	n_{1s}	a_1
X_2	n_{21}	n_{22}	$\cdots$	n_{2s}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
X_r	n_{r1}	n_{r2}	$\cdots$	n_{rs}	a_r
sums	b_1	b_2	$\cdots$	b_s	

$$ARI = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] \Big/ \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2} \sum_j \binom{b_j}{2}\right] \Big/ \binom{n}{2}}$$

  - This is based on the "permutation model" where the expectation is computed by randomly shuffling the samples among the clusters.

Objective Evaluation of Clustering

- **Normalized Mutual Information** (NMI):
  - The idea: Treat the two partitions as two discrete random variables, the **mutual information** is how much we can learn about one from the other. In other words, it is a measure of how much they are not independent.

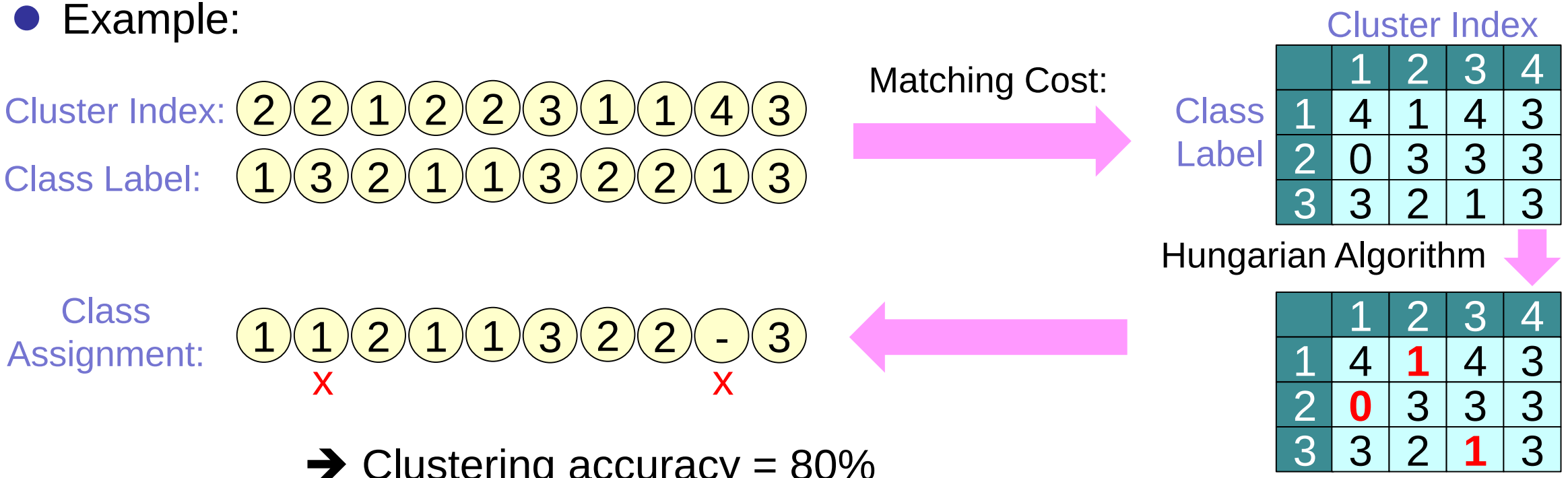
$$\mathrm{I}(X;Y) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} P_{(X,Y)}(x,y) \log \left(\frac{P_{(X,Y)}(x,y)}{P_X(x)\, P_Y(y)} \right)$$

  - For our problem, X and Y represent two different partitions, and x and y are cluster indices of the samples.
  - Normalized mutual information (normalized over the entropy of the two partitions themselves):

$$NMI = \frac{I(X;Y)}{\sqrt{H(X)H(Y)}}$$

Objective Evaluation of Clustering

- **Clustering Accuracy**:
  - The idea is to evaluate the clustering result just like a classification problem.
  - We need to first determine an optimal assignment between the class labels (ground truth) and the cluster indices of the data points. This is handled with the **Hungarian Algorithm**.
  - Example:



➔ Clustering accuracy = 80%