Data Mining (Computer Science Master's Degree)
Practice exercises

Lorenzo Mannocci

a.y. 2026/2027

Imbalanced data – Practice exercises

Instructions. Use Euclidean distance. When distances are tied, select the alphabetically first nearest neighbor.

Exercise 1 – Condensed Nearest Neighbor

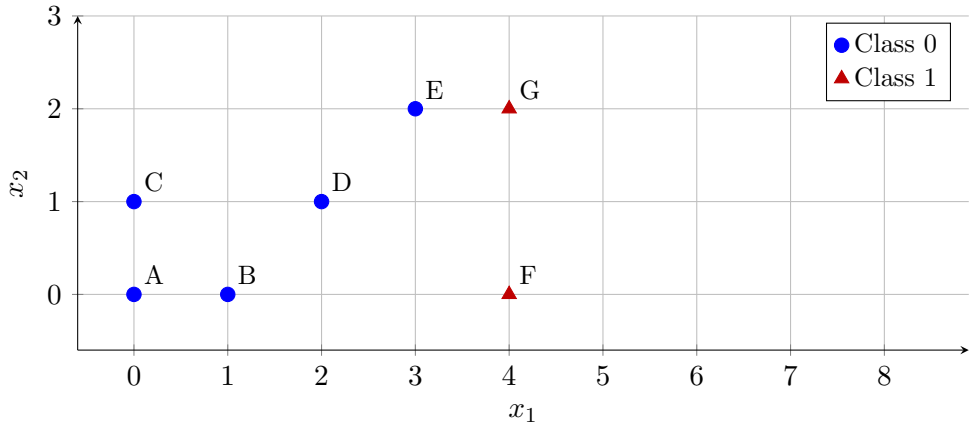
Consider the dataset below, with class 0 as the majority class and class 1 as the minority class.

Point	x_1	x_2	Class
A	0	0	0
B	1	0	0
C	0	1	0
D	2	1	0
E	3	2	0
F	4	0	1
G	4	2	1

Apply Condensed Nearest Neighbor (CNN) with a 1-nearest-neighbor classifier. The initial condensed set is $S = \{A, F, G\}$. Examine the remaining points in the order B, C, D, E.

Solution

When a point is misclassified, add it to S *immediately*; repeat full passes until a pass adds no points.



The 1-NN rule predicts the class of the nearest point *currently in* S. A correctly predicted point stays outside S; a misclassified point is added immediately.

First pass.

So, we have that $S = \{A, F, G\}$.

Pass	Point	Nearest in current S	Predicted class	True Class	Action
1	B	A ($d = 1$)	0	0	Do not add
1	C	A ($d = 1$)	0	0	Do not add
1	D	A ($d = \sqrt{5}$; tie with F and G)	0	0	Do not add
1	E	G ($d = 1$)	1	0	Add E

For D, $d(D, A) = d(D, F) = d(D, G) = \sqrt{5}$. The alphabetical tie-break selects A, which has class 0. This is why D is not added. For E, $d(E, G) = 1$, so E is misclassified as class 1 and is added.

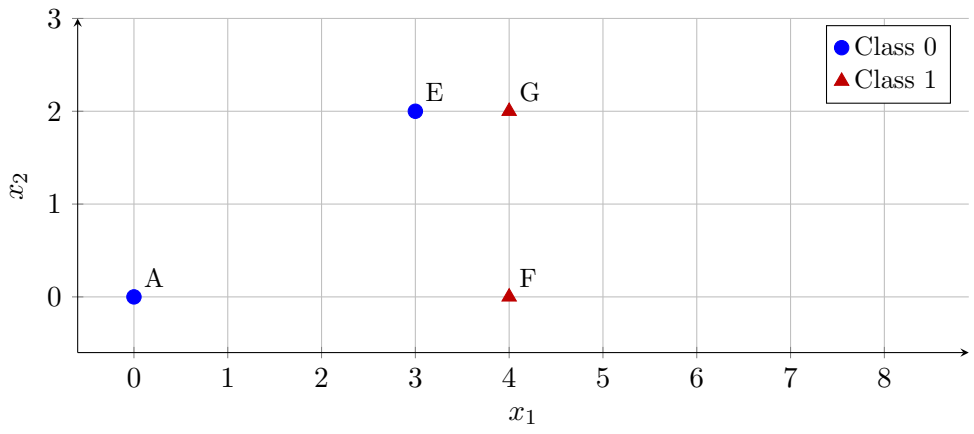
Update $S = \{A, F, G, E\}$.

Second pass.

Pass	Point	Nearest in current S	Predicted class	True Class	Action
2	B	A ($d = 1$)	0	0	Do not add
2	C	A ($d = 1$)	0	0	Do not add
2	D	E ($d = \sqrt{2}$)	0	0	Do not add

The second pass adds no new point (E is already in S), so the algorithm stops:

$$\boxed{S_{\text{final}} = \{A, E, F, G\}.}$$



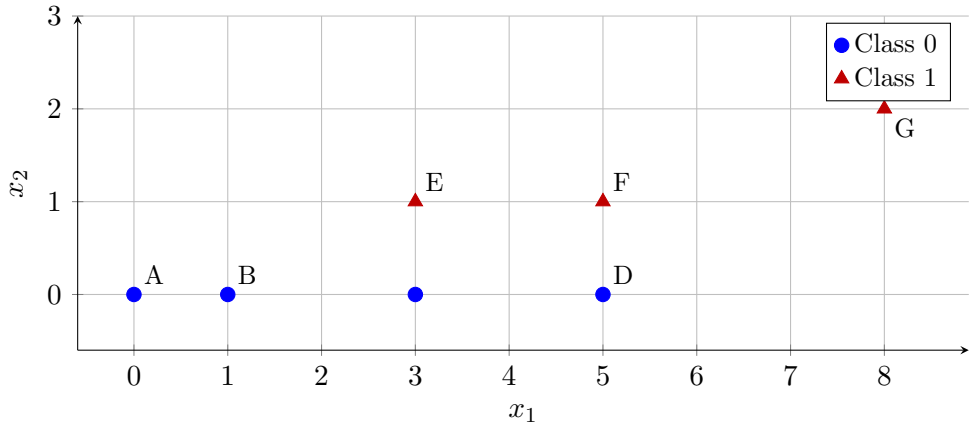
S_{final} is the undersampled dataset with the reduced number of point of the majority class.

Exercise 2 – Tomek Links

Consider the dataset below, where class 0 is selected for undersampling. Apply Tomek's links algorithm, by identifying all Tomek Links, and stating which observations are removed when only class 0 is undersampled.

Point	x_1	x_2	Class
A	0	0	0
B	1	0	0
C	3	0	0
D	5	0	0
E	3	1	1
F	5	1	1
G	8	2	1

Solution



A Tomek Link is a pair of opposite-class observations that are mutual nearest neighbors.

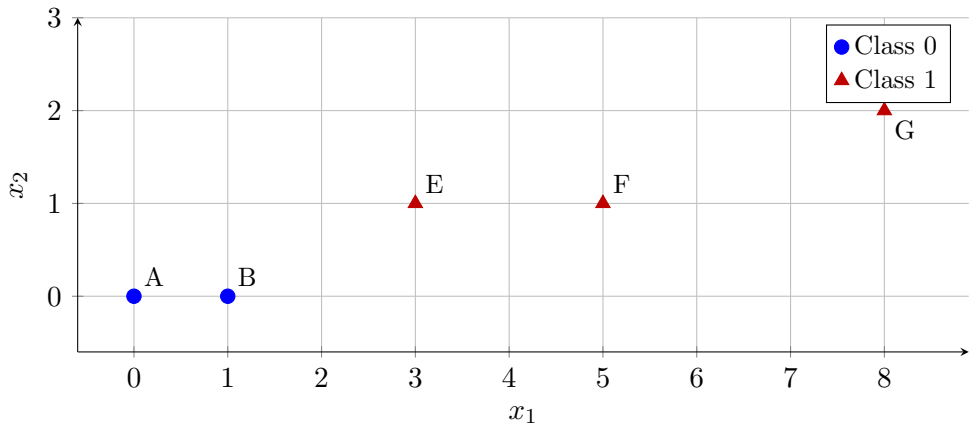
For each point, compare Euclidean distances to all other points, ignoring the point itself. The nearest neighbors are:

Point	Nearest neighbor	Distance
A	B	1
B	A	1
C	E	1
D	F	1
E	C	1
F	D	1
G	F	$\sqrt{10}$

A–B is mutual but has the same class, so it is *not* a Tomek Link. C–E and D–F are opposite-class mutual nearest-neighbor pairs:

$$\boxed{\text{Tomek Links: } (C, E),\ (D, F).}$$

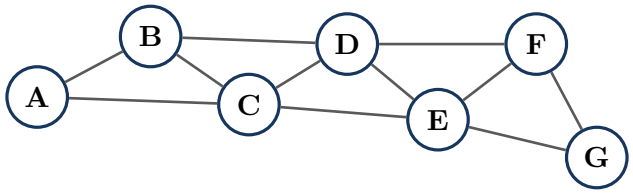
Since only class 0 is undersampled, remove C and D. The final dataset is $\{A, B, E, F, G\}$, with class counts 2 (class 0) and 3 (class 1). The method does not enforce class balance.


Class 0
Class 1
A
B
E
F
G
x_1
x_2

Network Science – Exercises

Exercise 1 – Basic network characterization

Consider the undirected graph below. All edges are unweighted.



a) Write the adjacency matrix in the node order A, B, C, D, E, F, G.

b) Compute the number of nodes N, edges L, each node's degree, and the average degree.

c) Compute density and the degree distribution $P(k)$.

d) Determine the graph diameter and justify it using shortest-path distances.

Solution

The graph has $N = 7$ nodes and $L = 11$ edges. In the specified order,

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}.$$

The degree sequence in node order is $(2, 3, 4, 4, 4, 3, 2)$; its sum is $22 = 2L$. Hence

$$\langle k \rangle = \frac{2L}{N} = \frac{22}{7} \approx 3.143, \qquad \rho = \frac{2L}{N(N-1)} = \frac{22}{42} = \frac{11}{21} \approx 0.524.$$

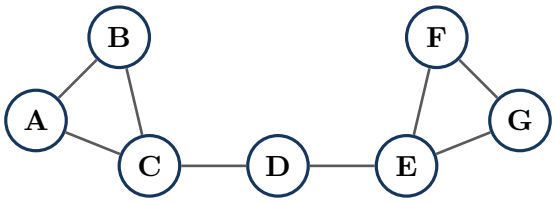
The empirical degree distribution is

$$P(2) = \frac{2}{7}, \qquad P(3) = \frac{2}{7}, \qquad P(4) = \frac{3}{7},$$

with $P(k) = 0$ otherwise. The diameter is $\boxed{3}$: for example, $d(A, G) = 3$ via $A - C - E - G$, and all other node pairs are at distance at most 3.

Exercise 2 – Clustering and centrality

Consider two tightly connected groups joined through node D.



a) Calculate the local clustering coefficients of C, D, and E, and the average clustering coefficient of the graph.

b) Calculate global clustering coefficient.

c) Compute the degree centrality of C and D.

Solution

a) Clustering coefficient and average clustering coefficient.

$$C_v = \begin{cases} \dfrac{2n_e(v)}{k_v(k_v-1)}, & k_v \geq 2, \\ 0, & k_v < 2. \end{cases}$$

where:

- k_v is the number of neighbors of node v.
- $n_e(v)$ is the number of edges between the neighbors of node v.

The degrees are $(2, 2, 3, 2, 3, 2, 2)$. The neighbors of C are $\{A, B, D\}$, with only edge AB among them; similarly, E has only FG among its three neighbors. Node D has neighbors $\{C, E\}$, which are not connected. Therefore

$$C_C = \frac{2 \cdot 1}{3 \cdot 2} = \frac{1}{3}, \qquad C_D = 0, \qquad C_E = \frac{1}{3}.$$

The four other nodes (A, B, F, G) each have clustering 1, so average clustering coefficient is:

$$\overline{C} = \frac{1+1+1+1+1/3+0+1/3}{7} = \boxed{\frac{2}{3}}.$$

b) Global clustering coefficient

For a node v with degree k_v, the number of connected triplets centered at v is:

$$\boxed{N_{\text{triplets}}(v) = \binom{k_v}{2} = \frac{k_v(k_v-1)}{2}}$$

Therefore, the total number of connected triplets in an undirected graph is:

$$\boxed{N_{\text{triplets}} = \sum_{v \in V} \binom{k_v}{2} = \frac{1}{2} \sum_{v \in V} k_v(k_v-1)}$$

In our graph, we obtain:

$$\begin{aligned} N_{\text{triplets}} &= \frac{1}{2}\sum_{v\in V} k_v(k_v-1) \\ &= \frac{1}{2}\left[2\cdot 1+2\cdot 1+3\cdot 2+2\cdot 1+3\cdot 2+2\cdot 1+2\cdot 1\right] \\ &= \frac{1}{2}(2+2+6+2+6+2+2) \\ &= \frac{1}{2}\cdot 22 \\ &= \boxed{11}. \end{aligned}$$

The global clustering coefficient measures the fraction of connected triplets that are closed, meaning that the two neighbors of the central node are also connected. It is defined as:

$$C_{\text{global}} = \frac{N_{\text{closed triplets}}}{N_{\text{triplets}}} = \frac{3N_{\text{triangles}}}{N_{\text{triplets}}}$$

since every triangle generates exactly three closed triplets (one centered at each vertex).

Hence in our graph is

$$C_{\text{global}} = \frac{3\times 2}{11} = \boxed{\frac{6}{11}}.$$

c) Degree centrality.

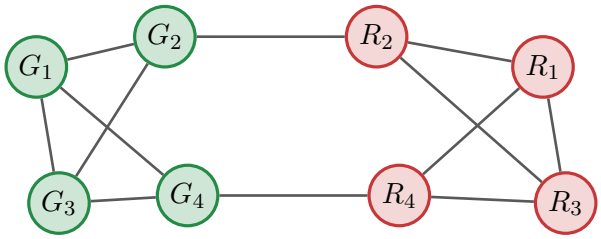
$$C_D(v) = \frac{k_v}{N-1}$$

- k_v is the degree of node v.
- N is the total number of nodes in the graph.

Normalized degree centralities are $C_D^{\text{deg}}(C) = 3/6 = 1/2$ and $C_D^{\text{deg}}(D) = 2/6 = 1/3$.

Exercise 3 – Categorical assortativity

Green and red nodes represent two categories. Compute Newman's categorical assortativity for the following undirected graph.



Solution

a) Count the number of green–green, red–red, and green–red edges.

b) Compute the quantities required by Newman's categorical assortativity formula.

c) Calculate the assortativity coefficient r and interpret its value.

Newman's categorical assortativity. The general formula for categorical assortativity is:

$$\boxed{r = \frac{\sum_i e_{ii} - \sum_i a_i b_i}{1 - \sum_i a_i b_i}}$$

where:

- e_{ij} is the fraction of links connecting nodes of type i to nodes of type j;
- a_i is the fraction of outgoing links from nodes of type i;
- b_i is the fraction of incoming links to nodes of type i.

Since the graph is undirected, there is no distinction between incoming and outgoing links. Therefore, $a_i = b_i$, and the formula simplifies to:

$$\boxed{r = \frac{\sum_i e_{ii} - \sum_i a_i^2}{1 - \sum_i a_i^2}}$$

Step 1: Count the edges. From the graph, we obtain:

$$L_{GG} = 5, \qquad L_{RR} = 5, \qquad L_{GR} = 2.$$

The total number of edges is:

$$L = L_{GG} + L_{RR} + L_{GR} = 5 + 5 + 2 = 12.$$

Step 2: Compute the fraction of links connecting nodes of the same category. The quantities e_{GG} and e_{RR} represent the fractions of links connecting nodes of the same category:

$$e_{GG} = \frac{L_{GG}}{L} = \frac{5}{12}$$

$$e_{RR} = \frac{L_{RR}}{L} = \frac{5}{12}$$

Therefore:

$$\sum_i e_{ii} = e_{GG} + e_{RR} = \frac{5}{12} + \frac{5}{12} = \frac{10}{12}.$$

Step 3: Compute a_G and a_R. For undirected graphs, a_i represents the fraction of edge endpoints belonging to nodes of category i.

Each undirected edge has two endpoints:

- A green–green edge contributes two green endpoints.
- A red–red edge contributes two red endpoints.
- A green–red edge contributes one green endpoint and one red endpoint.

Since the network has $L = 12$ edges, the total number of edge endpoints is $2L = 24$.

For the green category:

$$a_G = \frac{2L_{GG} + L_{GR}}{2L} = \frac{2 \cdot 5 + 2}{2 \cdot 12} = \frac{12}{24} = \frac{1}{2}.$$

For the red category:

$$a_R = \frac{2L_{RR} + L_{GR}}{2L} = \frac{2 \cdot 5 + 2}{2 \cdot 12} = \frac{12}{24} = \frac{1}{2}.$$

Notice that:

$$a_G + a_R = 1.$$

Since the graph is undirected:

$$b_G = a_G = \frac{1}{2}, \qquad b_R = a_R = \frac{1}{2}.$$

Step 4: Compute assortativity. We now substitute the computed quantities into the simplified formula:

$$r = \frac{\sum_i e_{ii} - \sum_i a_i^2}{1 - \sum_i a_i^2}.$$

Since there are only two categories, green and red:

$$r = \frac{(e_{GG} + e_{RR}) - (a_G^2 + a_R^2)}{1 - (a_G^2 + a_R^2)}.$$

Substituting the values:

$$
\begin{aligned}
r &= \frac{\left(\frac{5}{12}+\frac{5}{12}\right) - \left[\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right]}{1 - \left[\left(\frac{1}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right]} \\
&= \frac{\frac{10}{12} - \frac{1}{2}}{1 - \frac{1}{2}} \\
&= \frac{\frac{1}{3}}{\frac{1}{2}} \\
&= \boxed{\frac{2}{3} \approx 0.667}.
\end{aligned}
$$

Interpretation. The assortativity coefficient is positive ($r \approx 0.667$), indicating that nodes tend to connect preferentially to nodes belonging to the same category.

The network therefore exhibits positive categorical assortativity.

In particular, 10 out of 12 edges connect nodes of the same color. However, assortativity does not simply measure this proportion: it compares the observed fraction of same-category connections with the fraction expected under random mixing, taking into account the proportions of edge endpoints belonging to each category.

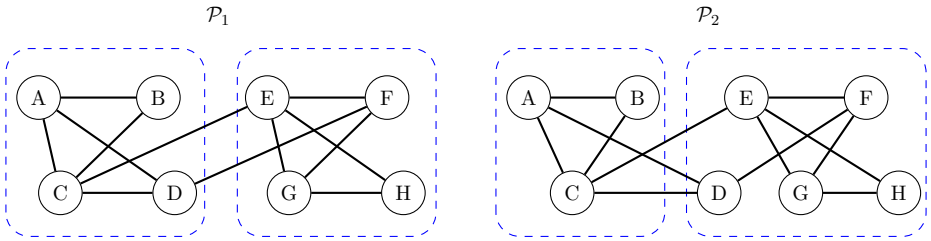
Exercise 4 – Modularity and community partitions

The graph below has 8 nodes and 12 edges. Consider the following two partitions:

$$\mathcal{P}_1 = \{\{A, B, C, D\}, \{E, F, G, H\}\}$$

$$\mathcal{P}_2 = \{\{A, B, C\}, \{D, E, F, G, H\}\}$$

The dashed outlines indicate the communities corresponding to each partition.



a) Compute modularity $Q(\mathcal{P}_1)$ and $Q(\mathcal{P}_2)$ using the community-level formula.

b) Which partition has higher modularity? Explain why the number of internal edges alone is not a sufficient criterion.

c) Explain briefly how modularity relates to Louvain's community-detection algorithm.

Solution

Modularity.

The modularity of a partition is defined as:

$$Q = \frac{1}{2L} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2L} \right) \delta(c_i, c_j),$$

where A_{ij} is the adjacency matrix, k_i and k_j are the degrees of nodes i and j, L is the total number of edges, and $\delta(c_i, c_j) = 1$ if nodes i and j belong to the same community, and 0 otherwise.

For an undirected, unweighted graph, this formula can be rewritten by grouping the terms belonging to the same community:

$$Q = \frac{1}{2L} \sum_{c} \sum_{i,j \in c} \left(A_{ij} - \frac{k_i k_j}{2L} \right).$$

We can consider the two terms separately.

First term: observed internal edges.

Since the graph is undirected, each internal edge is counted twice in the adjacency matrix:

$$\frac{1}{2L} \sum_{i,j \in c} A_{ij} = \frac{2L_c}{2L} = \frac{L_c}{L},$$

where L_c is the number of edges within community c.

Second term: expected internal edges.

$$\frac{1}{2L}\sum_{i,j\in c}\frac{k_i k_j}{2L} = \frac{1}{(2L)^2}\left(\sum_{i\in c}k_i\right)\left(\sum_{j\in c}k_j\right)$$
$$= \left(\frac{d_c}{2L}\right)^2,$$

where $d_c = \sum_{i\in c} k_i$ is the sum of the degrees of all nodes belonging to community c.

Combining the two terms, we obtain the equivalent community-level formulation:

$$\boxed{Q = \sum_c \left[\frac{L_c}{L} - \left(\frac{d_c}{2L}\right)^2\right]}$$

This formulation is more convenient for manual calculations, since it only requires counting the internal edges and summing the node degrees for each community.

Here $L = 12$, so $2L = 24$. For $\mathcal{P}_1$, both communities have $L_c = 5$ and $d_c = 12$:

$$Q(\mathcal{P}_1) = 2\left[\frac{5}{12} - \left(\frac{12}{24}\right)^2\right] = \boxed{\frac{1}{3} \approx 0.333}.$$

For $\mathcal{P}_2$, the community $\{A, B, C\}$ has $L_c = 3$ and $d_c = 9$; $\{D, E, F, G, H\}$ has $L_c = 6$ and $d_c = 15$:

$$Q(\mathcal{P}_2) = \left[\frac{3}{12} - \left(\frac{9}{24}\right)^2\right] + \left[\frac{6}{12} - \left(\frac{15}{24}\right)^2\right]$$
$$= \boxed{\frac{7}{32} = 0.21875}.$$

Thus, $\mathcal{P}_1$ is the better partition according to the modularity criterion, since it achieves a higher modularity ($Q(\mathcal{P}_1) > Q(\mathcal{P}_2)$).

Although $\mathcal{P}_1$ has more internal edges (10 versus 9), the number of internal edges alone is not sufficient to determine the quality of a partition. Modularity also accounts for the *expected* internal connectivity under a degree-based null model.

Therefore, a higher modularity indicates a better community partition according to this criterion, and community detection algorithms such as Louvain aim to *maximize modularity*. Louvain is a heuristic that repeatedly moves nodes between communities when doing so increases modularity, then aggregates communities into supernodes and repeats the process.

Clustering Heterogeneous and Multi-Source Data

Practice exercises with worked solutions

General instructions. When categorical modes are tied, choose the alphabetically first category. When two cluster representatives have equal dissimilarity, choose the cluster with the smaller index. All assignments are hard assignments. Unless specified otherwise, use squared Euclidean distance for numerical attributes and simple matching dissimilarity for categorical attributes.

Exercise 1 – K-Modes: Assignment and Mode Update

Consider the following dataset of six objects, each described by three categorical attributes.

Object	Color	Shape	Size
A	Red	Circle	Small
B	Red	Square	Small
C	Blue	Circle	Large
D	Blue	Square	Large
E	Red	Circle	Large
F	Blue	Circle	Small

Run one full assignment-and-update iteration of K-Modes with $k = 2$ and initial modes

$$Q_1^{(0)} = (\text{Red}, \text{Circle}, \text{Small}), \qquad Q_2^{(0)} = (\text{Blue}, \text{Square}, \text{Large}).$$

Solution

(a) Define simple matching dissimilarity and compute the dissimilarity of each object from both initial modes.

(b) Assign each object to a cluster, applying the specified tie-breaking rule.

(c) Update both modes using the most frequent category of each attribute.

(d) Compute the K-Modes objective using the updated modes and the assignments obtained in (b). Does another assignment step change any membership?

(a) Dissimilarities. For m categorical attributes,

$$d(X_i, Q_\ell) = \sum_{j=1}^{m} \delta(x_{ij}, q_{\ell j}), \qquad \delta(a, b) = \begin{cases} 0, & a = b, \\ 1, & a \neq b. \end{cases}$$

For instance, $B = (\text{Red}, \text{Square}, \text{Small})$ differs from $Q_1^{(0)}$ only in Shape, so $d(B, Q_1^{(0)}) = 1$; it differs from $Q_2^{(0)}$ in Color and Size, so $d(B, Q_2^{(0)}) = 2$.

Object	$d(X_i, Q_1^{(0)})$	$d(X_i, Q_2^{(0)})$	Assigned cluster
A	0	3	1
B	1	2	1
C	2	1	2
D	3	0	2
E	1	2	1
F	1	2	1

(b) Assignments. Every object is assigned to its nearest mode:

$$C_1 = \{A, B, E, F\}, \qquad C_2 = \{C, D\}.$$

(c) Mode update. For each cluster and attribute, select the most frequent category:

$$q_{\ell j}^{(1)} \in \arg\max_{a} \sum_{X_i \in C_\ell} \mathbf{1}(x_{ij} = a).$$

In C_1, Color is Red three times and Blue once; Shape is Circle three times and Square once; Size is Small three times and Large once. Hence

$$\boxed{Q_1^{(1)} = (\text{Red}, \text{Circle}, \text{Small}).}$$

In C_2, Color is Blue twice, Size is Large twice, and Shape is tied between Circle and Square. Alphabetical tie-breaking selects Circle:

$$\boxed{Q_2^{(1)} = (\text{Blue}, \text{Circle}, \text{Large}).}$$

(d) Objective and next assignment. For hard assignments, the objective is

$$P(W, Q) = \sum_{\ell=1}^{k} \sum_{i=1}^{n} w_{i\ell} d(X_i, Q_\ell) = \sum_{\ell=1}^{k} \sum_{X_i \in C_\ell} d(X_i, Q_\ell).$$

Using the updated modes, the within-cluster costs are

$$\begin{aligned} C_1: \quad & d(A, Q_1^{(1)}) + d(B, Q_1^{(1)}) + d(E, Q_1^{(1)}) + d(F, Q_1^{(1)}) \\ & = 0 + 1 + 1 + 1 = 3, \\ C_2: \quad & d(C, Q_2^{(1)}) + d(D, Q_2^{(1)}) = 0 + 1 = 1. \end{aligned}$$

Thus $\boxed{P(W, Q^{(1)}) = 4}$.

In a new assignment step, F has distance 1 from $Q_1^{(1)}$ and distance 1 from $Q_2^{(1)}$; the tie rule keeps it in cluster 1. All other objects also keep their assignments. Therefore the clustering is stable.

Exercise 2 – K-Prototypes: Mixed Dissimilarity and Updates

Consider six objects with two numerical attributes and one categorical attribute. Income is expressed in thousands of currency units. Apply K-prototypes algorithm.

Object	Age	Income	Job
A	20	20	Student
B	22	22	Student
C	24	24	Engineer
D	30	30	Engineer
E	32	32	Engineer
F	34	34	Teacher

Use $k = 2$, $\gamma = 8$, and the initial prototypes

$$Q_1^{(0)} = (22, 22, \text{Student}), \qquad Q_2^{(0)} = (32, 32, \text{Engineer}).$$

Solution

(a) Write the K-Prototypes dissimilarity and compute the distances to both prototypes for every object.

(b) Assign the objects and update the numerical means and categorical modes.

(c) Calculate the objective value using the updated prototypes and the assignments from (b).

(a) Mixed dissimilarity. With two numerical attributes and one categorical attribute,

$$d(X_i, Q_\ell) = (\text{Age}_i - \mu_{\ell,\text{Age}})^2 + (\text{Income}_i - \mu_{\ell,\text{Income}})^2 + \gamma\,\mathbf{1}(\text{Job}_i \neq \text{Job}_\ell).$$

For C and $Q_1^{(0)}$, $d(C, Q_1^{(0)}) = (24-22)^2 + (24-22)^2 + 8 = 16$. The full calculation is:

Object	$d(X_i, Q_1^{(0)})$	$d(X_i, Q_2^{(0)})$	Cluster
A	8	296	1
B	0	208	1
C	16	128	1
D	136	8	2
E	208	0	2
F	296	16	2

(b) Assignment and update. Thus

$$C_1 = \{A, B, C\}, \qquad C_2 = \{D, E, F\}.$$

The numerical prototype coordinates are means:

$$\mu_{\ell j} = \frac{1}{|C_\ell|} \sum_{X_i \in C_\ell} x_{ij},$$

and the categorical coordinate is the most frequent category. For C_1, the means are $(22, 22)$ and Student occurs twice, giving

$$\boxed{Q_1^{(1)} = (22, 22, \text{Student}).}$$

For C_2, the means are $(32, 32)$ and Engineer occurs twice, giving

$$\boxed{Q_2^{(1)} = (32, 32, \text{Engineer}).}$$

The prototypes are unchanged after this iteration.

(c) Objective. The total mixed within-cluster dissimilarity is

$$\begin{aligned} P(W, Q^{(1)}) &= (8 + 0 + 16) + (8 + 0 + 16) \\ &= \boxed{48}. \end{aligned}$$

Exercise 3 – Early Fusion: Distances and View Weights

Four objects have two numerical views, each with two features:

	View 1		*View 2*	
Object	$x_1^{(1)}$	$x_2^{(1)}$	$x_1^{(2)}$	$x_2^{(2)}$
A	0	0	0	0
B	1	1	3	4
C	3	0	1	0
D	3	2	4	4

(a) Write the early-fusion representation of A, B, and C by concatenating their two views.

(b) Calculate squared Euclidean distances $d^2(A, B)$ and $d^2(A, C)$ after concatenation, explicitly showing the contribution of each view. Which of B or C is nearer to A?

(c) Use the weighted early-fusion distance $d^2_\alpha(i, j) = \alpha_1 \|x_i^{(1)} - x_j^{(1)}\|^2 + \alpha_2 \|x_i^{(2)} - x_j^{(2)}\|^2$ with $(\alpha_1, \alpha_2) = (4, 0.2)$. Recompute both distances and identify the nearer object.

(d) Explain why per-feature standardization does not necessarily give equal importance to views with different numbers of features.

Solution

(a) Concatenation. Early fusion constructs $x_i = [x_i^{(1)} \mid x_i^{(2)}]$:

$$x_A = (0, 0, 0, 0), \qquad x_B = (1, 1, 3, 4), \qquad x_C = (3, 0, 1, 0).$$

(b) Unweighted distances. Squared Euclidean distance decomposes into the sum of the view-specific squared distances:

$$d^2(i, j) = \sum_{v=1}^{2} \|x_i^{(v)} - x_j^{(v)}\|_2^2.$$

For A and B, View 1 contributes $(1 - 0)^2 + (1 - 0)^2 = 2$, while View 2 contributes $3^2 + 4^2 = 25$; thus $d^2(A, B) = 2 + 25 = \boxed{27}$. For A and C, the contributions are $3^2 + 0^2 = 9$ and $1^2 + 0^2 = 1$; hence $d^2(A, C) = 9 + 1 = \boxed{10}$. Therefore C is nearer to A.

(c) Weighted distances.

$$d^2_\alpha(A, B) = 4(2) + 0.2(25) = \boxed{13},$$
$$d^2_\alpha(A, C) = 4(9) + 0.2(1) = \boxed{36.2}.$$

Now B is nearer to A. View weighting can therefore change the geometry and clustering result.

(d) Scaling versus view weighting. Standardization typically rescales each feature separately to unit variance. Euclidean distance still sums one squared term per feature, so a view with more standardized features may contribute more terms and dominate. Additional view-level weighting or normalization may be needed to balance contributions.

Exercise 4 – MCLA: Hypergraph and Meta-Graph Construction

Two independent clusterings of the same six objects $X = \{A, B, C, D, E, F\}$ produce the following partitions:

$$\begin{array}{lll} \text{Partition 1:} & e_1 = \{A, B, C\}, & e_2 = \{D, E, F\}, \\ \text{Partition 2:} & e_3 = \{A, B\}, & e_4 = \{C, D\}, \quad e_5 = \{E, F\}. \end{array}$$

Each base cluster is a hyperedge. In MCLA, each hyperedge becomes a vertex u_a of the weighted meta-graph G_M.

(a) State the numbers of hypergraph vertices and hyperedges, and construct the incidence matrix $H \in \{0, 1\}^{6\times 5}$, with rows ordered A–F and columns ordered $e_1, \ldots, e_5$.

(b) Calculate Jaccard similarity $J(e_1, e_3)$, $J(e_1, e_4)$, $J(e_2, e_4)$, and $J(e_2, e_5)$, exploiting the incidence matrix.

(c) Construct the complete weighted adjacency matrix W of the meta-graph, setting $W_{aa} = 0$ and omitting edges of zero similarity.

(d) Explain why there are no positive-weight edges between u_1 and u_2, or between any two of u_3, u_4, u_5.

Solution

(a) Hypergraph incidence matrix. There are $n = 6$ vertices (original objects) and $m = 2 + 3 = 5$ hyperedges (base clusters). By definition,

$$H_{ia} = \mathbf{1}(x_i \in e_a).$$

Therefore

$H =$

	e_1	e_2	e_3	e_4	e_5
A	1	0	1	0	0
B	1	0	1	0	0
C	1	0	0	1	0
D	0	1	0	1	0
E	0	1	0	0	1
F	0	1	0	0	1

(b) Jaccard similarities.

The Jaccard similarity between two hyperedges can be computed directly from their binary indicator vectors:

$$J(e_a, e_b) = \frac{|e_a \cap e_b|}{|e_a \cup e_b|} = \frac{\mathbf{h}_a^\top \mathbf{h}_b}{\|\mathbf{h}_a\|_2^2 + \|\mathbf{h}_b\|_2^2 - \mathbf{h}_a^\top \mathbf{h}_b},$$

where $\mathbf{h}_a$ is column a of H.

For hyperedges e_1 and e_3, the corresponding columns of H are:

$$\mathbf{h}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad \mathbf{h}_3 = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

First, compute their dot product:

$$\mathbf{h}_1^\top \mathbf{h}_3 = 1 \cdot 1 + 1 \cdot 1 + 1 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 = 2.$$

Then, compute their squared Euclidean norms:

$$\|\mathbf{h}_1\|_2^2 = 1^2 + 1^2 + 1^2 = 3, \qquad \|\mathbf{h}_3\|_2^2 = 1^2 + 1^2 = 2.$$

Therefore:

$$\boxed{J(e_1, e_3) = \frac{2}{3 + 2 - 2} = \frac{2}{3}.}$$

The dot product counts the objects belonging to both hyperedges, while the squared norms count the objects belonging to each hyperedge. Subtracting the dot product in the denominator avoids counting the shared objects twice.

The requested values are

$$\boxed{J(e_1,e_3)=\tfrac{2}{3},\quad J(e_1,e_4)=\tfrac{1}{4},\quad J(e_2,e_4)=\tfrac{1}{4},\quad J(e_2,e_5)=\tfrac{2}{3}.}$$

(c) Meta-graph adjacency matrix.

G_M has vertices $u_1,\dots,u_5$, with u_a representing e_a. An edge between distinct vertices has weight $W_{ab}=J(e_a,e_b)$ if the similarity is positive. The complete matrix is

$$\boxed{W=\begin{pmatrix} 0 & 0 & \frac{2}{3} & \frac{1}{4} & 0\\ 0 & 0 & 0 & \frac{1}{4} & \frac{2}{3}\\ \frac{2}{3} & 0 & 0 & 0 & 0\\ \frac{1}{4} & \frac{1}{4} & 0 & 0 & 0\\ 0 & \frac{2}{3} & 0 & 0 & 0 \end{pmatrix}.}$$

(d) Partite structure. Within a hard partition, distinct clusters are disjoint. Hence $e_1\cap e_2=\varnothing$ and e_3,e_4,e_5 are pairwise disjoint, giving Jaccard similarity zero. Thus positive edges can only connect hyperedges from different input partitions: this meta-graph is bipartite.

Exercise 5 – MCLA: Collapse and Compete for Objects

Continue with the five hyperedges and the incidence matrix H from Exercise 4. Assume the meta-graph partitioning step has already returned the following $k=2$ meta-clusters:

$$\mathcal{M}_1=\{e_1,e_3\},\qquad \mathcal{M}_2=\{e_2,e_4,e_5\}.$$

(a) Compute the association vectors $\mathbf{a}_1$ and $\mathbf{a}_2$ obtained by collapsing the hyperedges in each meta-cluster.

(b) Construct the association matrix $A\in[0,1]^{6\times 2}$, with rows ordered A–F.

(c) Apply the "Compete for Objects" rule to obtain the final consensus partition. If an object has equal association with both meta-clusters, assign it to meta-cluster 1.

(d) Explain why the association values are not necessarily probabilities summing to one across meta-clusters.

Solution

(a) Collapsing meta-clusters. For meta-cluster $\mathcal{M}_j$ the association vector is the mean of the indicator vectors of its hyperedges:

$$\mathbf{a}_j=\frac{1}{|\mathcal{M}_j|}\sum_{e_a\in\mathcal{M}_j}\mathbf{h}_a.$$

Using the columns of H from Exercise 4,

$$\mathbf{a}_1=\frac{\mathbf{h}_1+\mathbf{h}_3}{2}=\boxed{(1,1,\tfrac{1}{2},0,0,0)^\top},$$
$$\mathbf{a}_2=\frac{\mathbf{h}_2+\mathbf{h}_4+\mathbf{h}_5}{3}=\boxed{(0,0,\tfrac{1}{3},\tfrac{2}{3},\tfrac{2}{3},\tfrac{2}{3})^\top}.$$

For example, object C belongs to e_1 but not e_3, hence its association with $\mathcal{M}_1$ is $1/2$; it belongs to e_4 but neither e_2 nor e_5, hence its association with $\mathcal{M}_2$ is $1/3$.

(b) Association matrix.

Place the association vectors side by side:

$$A = [\mathbf{a}_1 \mid \mathbf{a}_2] = \begin{array}{c|cc} & \mathcal{M}_1 & \mathcal{M}_2 \\ \hline A & 1 & 0 \\ B & 1 & 0 \\ C & \frac{1}{2} & \frac{1}{3} \\ D & 0 & \frac{2}{3} \\ E & 0 & \frac{2}{3} \\ F & 0 & \frac{2}{3} \end{array}.$$

(c) Compete for Objects. Each object is assigned to the meta-cluster with maximum association:

$$\lambda^*(x_i) = \arg\max_{j \in \{1,2\}} A_{ij}.$$

Objects A, B, and C favor $\mathcal{M}_1$; D, E, and F favor $\mathcal{M}_2$. Thus

$$\boxed{C_1^* = \{A, B, C\}, \qquad C_2^* = \{D, E, F\}.}$$

(d) Interpretation. A_{ij} is the fraction of base hyperedges within meta-cluster j that contain object i. Different meta-clusters can contain different numbers of hyperedges, and these fractions are computed independently. Hence $\sum_j A_{ij}$ need not equal 1; the entries are association strengths, not automatically normalized posterior probabilities.

Exercise 6 – Multi-View K-Means: Alternating Updates

The same six objects have two one-dimensional numerical views. We want $k = 2$ clusters.

Object	View 1: $x_i^{(1)}$	View 2: $x_i^{(2)}$
A	0	0
B	1	1
C	2	6
D	7	4
E	8	8
F	9	9

Suppose the current assignments obtained in View 1 are

$$C_1^{(1)} = \{A, B, C\}, \qquad C_2^{(1)} = \{D, E, F\}.$$

Perform one alternating cross-view update of Multi-View K-Means (do not run K-Means to convergence within either view):

(a) Transfer the View 1 memberships to View 2 and compute the two centroids using *View 2 coordinates*.

(b) Reassign every object to its nearest View 2 centroid. Show the squared distances for C and D explicitly.

(c) Transfer these updated memberships back to View 1 and compute the two new View 1 centroids.

(d) Reassign every object in View 1. State the resulting memberships and explain what information was transferred between views.

Solution

(a) Centroids in View 2 from View 1 assignments. With the transferred memberships, compute means in the destination view:

$$\mu_c^{(2)} = \frac{1}{|C_c^{(1)}|} \sum_{i \in C_c^{(1)}} x_i^{(2)}.$$

Therefore

$$\boxed{\mu_1^{(2)} = \frac{0+1+6}{3} = \frac{7}{3}}, \qquad \boxed{\mu_2^{(2)} = \frac{4+8+9}{3} = 7}.$$

(b) Reassign in View 2. The assignment rule is

$$c_i^{(2)} = \arg\min_{c \in \{1,2\}} (x_i^{(2)} - \mu_c^{(2)})^2.$$

For C ($x_C^{(2)} = 6$), the squared distances are

$$(6 - \tfrac{7}{3})^2 = \tfrac{121}{9}, \qquad (6-7)^2 = 1,$$

so C moves to cluster 2. For D ($x_D^{(2)} = 4$),

$$(4 - \tfrac{7}{3})^2 = \tfrac{25}{9}, \qquad (4-7)^2 = 9,$$

so D moves to cluster 1. The full assignment is

Object	View 2 value	d^2 to $\mu_1^{(2)}$	d^2 to $\mu_2^{(2)}$
A	0	49/9	49
B	1	16/9	36
C	6	121/9	1
D	4	25/9	9
E	8	289/9	1
F	9	400/9	4

Hence

$$\boxed{C_1^{(2)} = \{A, B, D\}, \qquad C_2^{(2)} = \{C, E, F\}.}$$

(c) Update View 1 using View 2 assignments. Recompute centroids in the *View 1 feature space*:

$$\mu_c^{(1)} = \frac{1}{|C_c^{(2)}|} \sum_{i \in C_c^{(2)}} x_i^{(1)}.$$

Thus

$$\boxed{\mu_1^{(1)} = \frac{0+1+7}{3} = \frac{8}{3}}, \qquad \boxed{\mu_2^{(1)} = \frac{2+8+9}{3} = \frac{19}{3}}.$$

(d) Reassign in View 1. The midpoint between the centroids is $(\frac{8}{3} + \frac{19}{3})/2 = \frac{9}{2} = 4.5$. Since squared distance to the nearer centroid is smaller, values below 4.5 go to cluster 1 and values above 4.5 go to cluster 2. Therefore

$$\boxed{C_1^{(1),\text{new}} = \{A, B, C\}, \qquad C_2^{(1),\text{new}} = \{D, E, F\}.}$$

The updated View 1 assignments happen to return to the initial partition. This does not mean that the View 2 assignment step was redundant: C and D switched memberships in View 2. The algorithm exchanges *object-to-cluster memberships*, not numerical centroid coordinates; each destination view computes its own centroids using its own features. One alternating update is not sufficient to assert convergence.