CSCI3230 (ESTR3108)
Fundamentals of Artificial Intelligence

Tutorial 5

Yuehao Wang

Email: yhwang21@cse.cuhk.edu.hk
Office: Room 1024, 10/F, SHB

Dept. of Computer Science & Engineering
The Chinese University of Hong Kong

Outline

Part 1. K-means

Part 2. DBSCAN

Part 3. Spectral clustering

Part 1. K-means

K-means algorithm

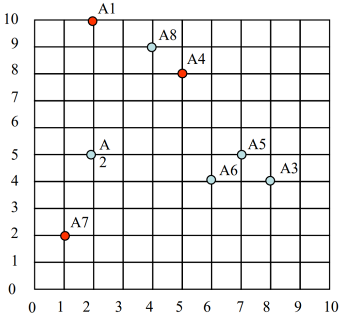
Review the routine of K-means:

1. Choose K (random) data points (as seeds) to be the initial **centroids** as cluster centers.
2. Assign each data point to the closest centroid.
3. Re-compute the centroids using the current cluster memberships.
4. If a convergence criterion is not met, repeat steps 2 and 3.

K-means exercise

Use the k-means algorithm and Euclidean distance.

- Data points: $A_1(2,10)$, $A_2(2,5)$, $A_3(8,4)$, $A_4(5,8)$, $A_5(7,5)$, $A_6(6,4)$, $A_7(1,2)$, $A_8(4,9)$
- Question: Please show the centroids of clusters after the 1st iteration of k-means. Suppose we set 3 clusters with initial centroids $u_1(2,10)$. $u_2(5,8)$, $u_3(1,2)$.



K-means exercise

Data points: $A_1(2,10)$, $A_2(2,5)$, $A_3(8,4)$, $A_4(5,8)$, $A_5(7,5)$, $A_6(6,4)$, $A_7(1,2)$, $A_8(4,9)$

	$u_1(2,10)$	$u_2(5,8)$	$u_3(1,2)$	class
A_1	0	$\sqrt{13}$	$\sqrt{65}$	1

K-means exercise

Data points: $A_1(2, 10)$, $A_2(2, 5)$, $A_3(8, 4)$, $A_4(5, 8)$, $A_5(7, 5)$, $A_6(6, 4)$, $A_7(1, 2)$, $A_8(4, 9)$

	$u_1(2, 10)$	$u_2(5, 8)$	$u_3(1, 2)$	class
A_1	0	$\sqrt{13}$	$\sqrt{65}$	1
A_2	5	$\sqrt{18}$	$\sqrt{10}$	3

K-means exercise

Data points: $A_1(2,10)$, $A_2(2,5)$, $A_3(8,4)$, $A_4(5,8)$, $A_5(7,5)$, $A_6(6,4)$, $A_7(1,2)$, $A_8(4,9)$

	$u_1(2,10)$	$u_2(5,8)$	$u_3(1,2)$	class
A_1	0	$\sqrt{13}$	$\sqrt{65}$	1
A_2	5	$\sqrt{18}$	$\sqrt{10}$	3
A_3	6	5	$\sqrt{53}$	2

K-means exercise

Data points: $A_1(2,10)$, $A_2(2,5)$, $A_3(8,4)$, $A_4(5,8)$, $A_5(7,5)$, $A_6(6,4)$, $A_7(1,2)$, $A_8(4,9)$

	$u_1(2,10)$	$u_2(5,8)$	$u_3(1,2)$	class
A_1	0	$\sqrt{13}$	$\sqrt{65}$	1
A_2	5	$\sqrt{18}$	$\sqrt{10}$	3
A_3	6	5	$\sqrt{53}$	2
A_4	$\sqrt{13}$	0	$\sqrt{50}$	2

K-means exercise

Data points: $A_1(2,10)$, $A_2(2,5)$, $A_3(8,4)$, $A_4(5,8)$, $A_5(7,5)$, $A_6(6,4)$, $A_7(1,2)$, $A_8(4,9)$

	$u_1(2,10)$	$u_2(5,8)$	$u_3(1,2)$	class
A_1	0	$\sqrt{13}$	$\sqrt{65}$	1
A_2	5	$\sqrt{18}$	$\sqrt{10}$	3
A_3	6	5	$\sqrt{53}$	2
A_4	$\sqrt{13}$	0	$\sqrt{50}$	2
A_5	$\sqrt{50}$	$\sqrt{13}$	$\sqrt{45}$	2

K-means exercise

Data points: $A_1(2, 10)$, $A_2(2, 5)$, $A_3(8, 4)$, $A_4(5, 8)$, $A_5(7, 5)$, $A_6(6, 4)$, $A_7(1, 2)$, $A_8(4, 9)$

	$u_1(2, 10)$	$u_2(5, 8)$	$u_3(1, 2)$	class
A_1	0	$\sqrt{13}$	$\sqrt{65}$	1
A_2	5	$\sqrt{18}$	$\sqrt{10}$	3
A_3	6	5	$\sqrt{53}$	2
A_4	$\sqrt{13}$	0	$\sqrt{50}$	2
A_5	$\sqrt{50}$	$\sqrt{13}$	$\sqrt{45}$	2
A_6	$\sqrt{52}$	$\sqrt{17}$	$\sqrt{29}$	2

K-means exercise

Data points: $A_1(2,10)$, $A_2(2,5)$, $A_3(8,4)$, $A_4(5,8)$, $A_5(7,5)$, $A_6(6,4)$, $A_7(1,2)$, $A_8(4,9)$

	$u_1(2,10)$	$u_2(5,8)$	$u_3(1,2)$	class
A_1	0	$\sqrt{13}$	$\sqrt{65}$	1
A_2	5	$\sqrt{18}$	$\sqrt{10}$	3
A_3	6	5	$\sqrt{53}$	2
A_4	$\sqrt{13}$	0	$\sqrt{50}$	2
A_5	$\sqrt{50}$	$\sqrt{13}$	$\sqrt{45}$	2
A_6	$\sqrt{52}$	$\sqrt{17}$	$\sqrt{29}$	2
A_7	$\sqrt{65}$	$\sqrt{52}$	0	3

K-means exercise

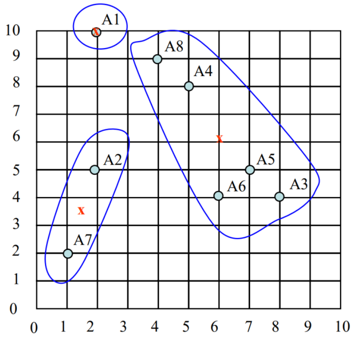
Data points: $A_1(2,10)$, $A_2(2,5)$, $A_3(8,4)$, $A_4(5,8)$, $A_5(7,5)$, $A_6(6,4)$, $A_7(1,2)$, $A_8(4,9)$

	$u_1(2,10)$	$u_2(5,8)$	$u_3(1,2)$	class
A_1	0	$\sqrt{13}$	$\sqrt{65}$	1
A_2	5	$\sqrt{18}$	$\sqrt{10}$	3
A_3	6	5	$\sqrt{53}$	2
A_4	$\sqrt{13}$	0	$\sqrt{50}$	2
A_5	$\sqrt{50}$	$\sqrt{13}$	$\sqrt{45}$	2
A_6	$\sqrt{52}$	$\sqrt{17}$	$\sqrt{29}$	2
A_7	$\sqrt{65}$	$\sqrt{52}$	0	3
A_8	$\sqrt{5}$	$\sqrt{2}$	$\sqrt{58}$	2

K-means exercise

Re-compute centroids based on the current clustering.

- Data points: $A_1(2,10)$, $A_2(2,5)$, $A_3(8,4)$, $A_4(5,8)$, $A_5(7,5)$, $A_6(6,4)$, $A_7(1,2)$, $A_8(4,9)$
- Cluster 1: $u_1 \leftarrow A_1 = (2,10)$
- Cluster 2: $u_2 \leftarrow (A_3 + A_4 + A_5 + A_6 + A_8)/5 = (6,6)$
- Cluster 3: $u_3 \leftarrow (A_2 + A_7)/2 = (1.5, 3.5)$

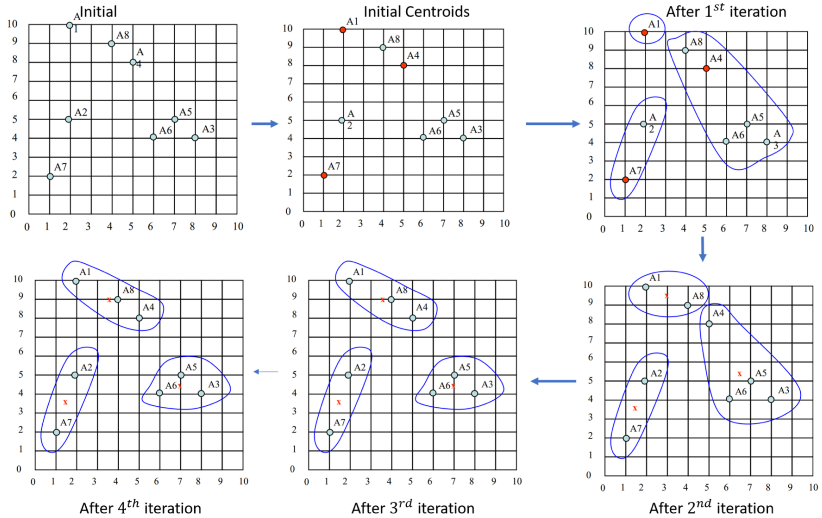


K-means exercise

Continue performing k-means until termination.

- 1st iteration, $\text{Cluster1} = \{A_1\}$, $\text{Cluster2} = \{A_3, A_4, A_5, A_6, A_8\}$. $\text{Cluster3} = \{A_2, A_7\}$ $u_1 = (2, 10), u_2 = (6, 6), u_3 = (1.5, 3.5)$
- 2nd iteration, $\text{Cluster1} = \{A_1, A_8\}$, $\text{Cluster2} = \{A_3, A_4, A_5, A_6, \}$. $\text{Cluster3} = \{A_2, A_7\}$ $u_1 = (3, 9.5), u_2 = (6.5, 5.25), u_3 = (1.5, 3.5)$
- 3rd iteration, $\text{Cluster1} = \{A_1, A_4, A_8\}$, $\text{Cluster2} = \{A_3, A_5, A_6, \}$. $\text{Cluster3} = \{A_2, A_7\}$ $u_1 = (3.66, 9), u_2 = (7, 4.33), u_3 = (1.5, 3.5)$
- 4th iteration, $\text{Cluster1} = \{A_1, A_4, A_8\}$, $\text{Cluster2} = \{A_3, A_5, A_6, \}$. $\text{Cluster3} = \{A_2, A_7\}$ $u_1 = (3.66, 9), u_2 = (7, 4.33), u_3 = (1.5, 3.5)$
- 5th iteration, $\text{Cluster1} = \{A_1, A_4, A_8\}$, $\text{Cluster2} = \{A_3, A_5, A_6, \}$. $\text{Cluster3} = \{A_2, A_7\}$ $u_1 = (3.66, 9), u_2 = (7, 4.33), u_3 = (1.5, 3.5)$
- ... converged after 3 iterations

K-means exercise



Part 2. DBSCAN

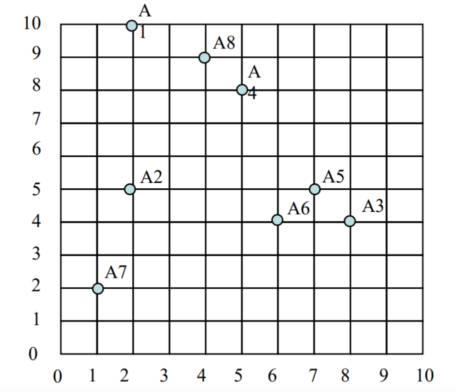
DBSCAN algorithm

- Determine core points and noise points.
- Starting from core points, BFS to find dense neighboring regions.

```
DBSCAN(DB, distFunc, eps, minPts) {
    C := 0                                                  /* Cluster counter */
    for each point P in database DB {
        if label(P) ≠ undefined then continue               /* Previously processed in inner loop */
        Neighbors N := RangeQuery(DB, distFunc, P, eps)     /* Find neighbors */
        if |N| < minPts then {                              /* Density check */
            label(P) := Noise                               /* Label as Noise */
            continue
        }
        C := C + 1                                          /* next cluster label */
        label(P) := C                                       /* Label initial point */
        SeedSet S := N \ {P}                                /* Neighbors to expand */
        for each point Q in S {                             /* Process every seed point Q */
            if label(Q) = Noise then label(Q) := C          /* Change Noise to border point */
            if label(Q) ≠ undefined then continue           /* Previously processed (e.g., border point) */
            label(Q) := C                                   /* Label neighbor */
            Neighbors N := RangeQuery(DB, distFunc, Q, eps) /* Find neighbors */
            if |N| ≥ minPts then {                          /* Density check (if Q is a core point) */
                S := S ∪ N                                  /* Add new neighbors to seed set */
            }
        }
    }
}
```

DBSCAN exercise

- Data points: $A_1(2,10)$, $A_2(2,5)$, $A_3(8,4)$, $A_4(5,8)$, $A_5(7,5)$, $A_6(6,4)$, $A_7(1,2)$, $A_8(4,9)$
- Let $\epsilon = \sqrt{5}, minPts = 3$.

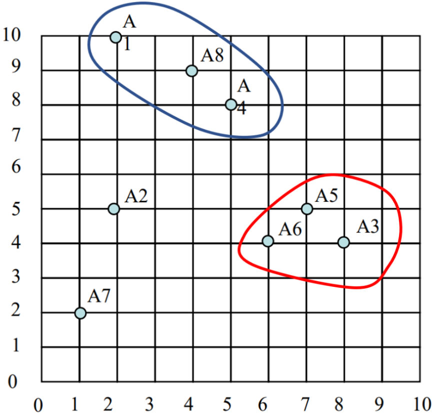


DBSCAN exercise

- Core points: A_3, A_5, A_6, A_8
- Border points: A_1, A_4,
- Noise points: A_2, A_7.

ε minPt

$\sqrt{5}$, 3



Part 3. Spectral clustering

Problem definition

Given a population of N objects, we want to divide the population into c classes, where c is unknown.

- We can introduce a similarity metric to measure the distance between each pair of objects.
- Similarity measurements: cosine distance $x \cdot y$, L2-norm $\|x - y\|_2$.
- By comparing similarities, we can obtain a non-negative symmetric affinity matrix W. W_{ij} is the similarity between the i-th object and j-th object.
- Sometimes we can threshold the similarity measurements: low similarities are rounded to 0.
- Then W would be a sparse matrix.

Problem definition

- Represent each object as a node.
- We can create a relational graph G based on the similarities. If the similarity between two nodes is non-zero, we will build an edge between them.
- W is the adjacency matrix of the relational graph.
- Let $G_1, \ldots, G_c$ be subgraphs of G corresponding to c clusters
- Idealized assumption: $G_j \cap G_i = \emptyset$ for any $i \neq j$. G_i's are connected.
- The problem can be converted to finding connected components in G.

Graph Laplacian

- The key object in this problem is the graph Laplacian.
- Graph Laplacian is defined as $L = D - W$, where D is the degree matrix $D_{ii} = \sum_{j=[N]} W_{ij}$.
- Properties of graph Laplacian:
  - Symmetric.
  - Positive semi-definite.
  - Not invertible.
  - The vector of all ones lies in its nullspace.

Spectral clustering

Denote the indicator vector of G_i by $e_{G_i} \in \mathbb{R}^N$, i.e. the vector that has ones at all coordinates indexed by the vertices of G_i and 0's elsewhere.

Theorem

$\dim N(L) = c$ *and a basis for* $N(L)$ *is given by the* e_{G_i}*'s.*

Spectral clustering

Theorem

Let $\xi_1, \dots, \xi_c \in \mathbb{R}^N$ be a basis for $N(L)$ and consider the matrix $Y = [y_1 \dots y_N] := [\xi_1 \dots \xi_c]^T \in \mathbb{R}^{c \times N}$. Then node $i, j \in [N]$ lie in the same connected component of G iff $y_i = y_j$.

This defines an embedding of the graph into the Euclidean space $\mathbb{R}^c$. Remarkably, the embedding takes nodes in the same cluster to the same vector of $\mathbb{R}^c$ with different vectors for different clusters, thus directly revealing the clustering label of each node.

References

- Von Luxburg, Ulrike. "A tutorial on spectral clustering." Statistics and computing 17.4 (2007): 395-416.
- Manolis C. Tsakiris,. "Lecture notes on linear algebra and applications." (2020).