Pattern Classification

Lecture 09: Clustering

Kundan Kumar
https://github.com/erkundanec/PatternClassification

© 2020 Kundan Kumar, All Rights Reserved

Unsupervised Learning: Clustering

Introduction

- In previous lectures, we have seen that how samples are classified if a training set is available along with class labels to design a classifier.
- In many situations, class labels are not known for the training samples.
- The area of pattern classification which cluster / group the data samples based on some similarity is known as clustering.
- Clustering refers to the process of grouping samples so that the samples are similar within each group. The groups are called *clusters*.
- There are two major issue in clustering:
  - how to measure similarity.
  - the criterion function to be optimized.

Hierarchical Clustering

- Hierarchical clustering refers to a clustering process that organizes the data into large groups, which contain smaller groups, and so on.
- Hierarchical clustering approach can be categorize as
  - Bottom-Up approach (called *Agglomerative* clustering)
  - Top-Down approach (called *Divisive* clustering)
- A hierarchical clustering may be drawn as a *tree* or *dendrogram*.

Agglomerative Clustering

- Consider $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n$ are n d-dimensional feature vectors.
- Algorithm:
  1: Begin with n clusters, each consisting of one sample.
  2: Repeat step 3 a total of $n-1$ times
  3: Find the most similar clusters C_i and C_j and merge C_i and C_j into one cluster. If there is a tie, merge the first pair found.

Agglomerative Clustering

- Agglomerative clustering can be performed using these approach:
  - Single-linkage algorithm

$$D_{SL}(C_i, C_j) = \min_{a \in C_i, b \in C_j} d(a, b)$$

  - Complete-linkage algorithm

$$D_{SL}(C_i, C_j) = \max_{a \in C_i, b \in C_j} d(a, b)$$

  - Average-linkage algorithm

$$D_{SL}(C_i, C_j) = \operatorname*{avg}_{a \in C_i, b \in C_j} d(a, b)$$

Example: Single-linkage algorithm

Question 01: Perform a hierarchical clustering of five samples using the single-linkage algorithm and two features x and y.

	x	y
1	4	4
2	8	4
3	15	8
4	24	4
5	24	12

	1	2	3	4	5
1	—	4.0	11.7	20.0	21.5
2	4.0	—	8.1	16.0	17.9
3	11.7	8.1	—	9.8	9.8
4	20.0	16.0	9.8	—	8.0
5	21.5	17.9	9.8	8.0	—

- Combine 1 and 2 in single cluster

$$\{1,2\},\{3\},\{4\},\{5\}$$

Example: Single-linkage algorithm

	{1,2}	3	4	5
{1,2}	—	8.1	16.0	17.9
3	8.1	—	9.8	9.8
4	16.0	9.8	—	8.0
5	17.9	9.8	8.0	—

- Minimum value is 8.0 so merge cluster $\{4\}$ and $\{5\}$.

$$\{1,2\},\{3\},\{4,5\}$$

Example: Single-linkage algorithm

	{1,2}	3	{4,5}
{1,2}	—	8.1	16.0
3	8.1	—	9.8
{4,5}	16.0	9.8	—

- Minimum value is 8.1 so merge cluster $\{1,2\}$ and $\{3\}$.

$$\{1,2,3\},\{4,5\}$$

- In next step will merge the two remaining clusters at a distance of 9.8.

Figure: Dendrogram of Hierarchical clustering using single-linkage algorithm

Example: Complete-linkage algorithm

Question 02: Perform a hierarchical clustering of five samples using the complete-linkage algorithm and two features x and y.

	x	y
1	4	4
2	8	4
3	15	8
4	24	4
5	24	12

	1	2	3	4	5
1	—	4.0	11.7	20.0	21.5
2	4.0	—	8.1	16.0	17.9
3	11.7	8.1	—	9.8	9.8
4	20.0	16.0	9.8	—	8.0
5	21.5	17.9	9.8	8.0	—

- Combine 1 and 2 in single cluster

$$\{1,2\},\{3\},\{4\},\{5\}$$

Example: Complete-linkage algorithm

	{1,2}	3	4	5
{1,2}	—	11.7	20.0	21.5
3	11.7	—	9.8	9.8
4	20.0	9.8	—	8.0
5	21.5	9.8	8.0	—

- Minimum value is 8.0 so merge cluster $\{4\}$ and $\{5\}$.

$$\{1,2\},\{3\},\{4,5\}$$

Example: Complete-linkage algorithm

	{1,2}	3	{4,5}
{1,2}	—	11.7	21.5
3	11.7	—	9.8
{4,5}	21.5	9.8	—

- Minimum value is 9.8 so merge cluster $\{4,5\}$ and $\{3\}$.

$$\{1,2\},\{3,4,5\}$$

- In next step will merge the two remaining clusters.

Figure: Dendrogram of Hierarchical clustering using complete-linkage algorithm

Example: Average-linkage algorithm

Question 03: Perform a hierarchical clustering of five samples using the average-linkage algorithm and two features x and y.

	x	y
1	4	4
2	8	4
3	15	8
4	24	4
5	24	12

	1	2	3	4	5
1	—	4.0	11.7	20.0	21.5
2	4.0	—	8.1	16.0	17.9
3	11.7	8.1	—	9.8	9.8
4	20.0	16.0	9.8	—	8.0
5	21.5	17.9	9.8	8.0	—

- Combine 1 and 2 in single cluster

$$\{1,2\},\{3\},\{4\},\{5\}$$

Example: Average-linkage algorithm

	{1,2}	3	4	5
{1,2}	—	9.9	18.0	19.7
3	9.9	—	9.8	9.8
4	18	9.8	—	8.0
5	19.7	9.8	8.0	—

.

- Minimum value is 8.0 so merge cluster $\{4\}$ and $\{5\}$.

$$\{1,2\},\{3\},\{4,5\}$$

Example: Average-linkage algorithm

	{1,2}	3	{4,5}
{1,2}	—	9.9	18.9
3	9.9	—	9.8
{4,5}	18.9	9.8	—

- Minimum value is 9.8 so merge cluster $\{4,5\}$ and $\{3\}$.

$$\{1,2\},\{3,4,5\}$$

- In next step will merge the two remaining clusters.

Figure: Dendrogram of Hierarchical clustering using average-linkage algorithm

Ward's Algorithm

- The hierarchical clustering based on variance.
- also called minimum-variance method
- Consider C_j class has n no. of feature vectors $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n$
- Error within j^{th} cluster

$$E_j = \sum_{i=1}^{n} \|\mathrm{x}_i - \mu\|^2 = m\sigma^2$$

  we need to minimize the variance
- Total error

$$E = \sum_{i=1}^{c} E_j$$

- Computationally expensive because need to check all combination of samples.

Example: Ward's algorithm

	x	y
1	4	4
2	8	4
3	15	8
4	24	4
5	24	12

Clusters	Squared Error, E
{1,2},{3},{4},{5}	8.0
{1,3},{2},{4},{5}	68.5
{1,4},{2},{3},{5}	200.0
{1,5},{2},{3},{4}	232.0
{2,3},{1},{4},{5}	32.5
{2,4},{1},{3},{5}	128.0
{2,5},{1},{3},{4}	160.0
{3,4},{1},{2},{5}	48.5
{3,5},{1},{2},{4}	48.5
{4,5},{1},{2},{3}	32.0

- Minimum Squared Error is 8.0 so form the cluster $\{1,2\},\{3\},\{4\},\{5\}$

Example: Ward's algorithm

Clusters	Squared Error, E
{1,2,3},{4},{5}	72.7
{1,2,4},{3},{5}	224.0
{1,2,5},{3},{4}	266.7
{1,2},{3,4},{5}	56.5
{1,2},{3,5},{4}	56.5
{1,2},{4,5},{3}	40.0

- Minimum Squared Error is 40.0 so form the cluster $\{1,2\},\{3\},\{4,5\}$

Example: Ward's algorithm

Clusters	Squared Error, E
{1,2,3},{4,5}	104.7
{1,2,4,5},{3}	380.0
{1,2},{3,4,5}	94.0

- Minimum value is 94.0 so form the cluster $\{1,2\},\{3,4,5\}$
- In next step will merge the two remaining clusters.

Figure: Dendrogram of Ward's algorithm

Batchelor and Wilkins' Algorithm

- Heuristic procedure for clustering also called *maximum distance algorithm*

Batchelor and Wilkins' Algorithm

Step 1: Arbitrarily, let x_1 be the first cluster center, designated by z_1.

Step 2: Determine the pattern sample farthest from x_1, which is x_6. Call it cluster center z_6.

Step 3: Compute the distance from each remaining pattern sample to z_1 and z_2

Step 4: Save the minimum distance for each pair of these computations.

Step 5: Select the maximum of these minimum distances.

Step 6: If the distance is appreciably greater than a fraction of the distance $d(\mathrm{z}_1, \mathrm{z}_2)$, call the corresponding sample cluster center z_3. Otherwise, the algorithm is terminated.

Step 7: If this distance from each of the three established cluster centers to the remaining samples and save the minimum of every group of three distances. Again select the maximum of these minimum distances. If this distance is an appreciable fraction of the "typical" previous maximum distances, the corresponding sample becomes cluster center z_4. Otherwise, the algorithm is terminated.

Step 8: Repeat until the new maximum distance at a particular step fails to satisfy the condition for the creation of a new cluster center.

Step 9: Assign each sample to its nearest cluster center.

Batchelor and Wilkins' Algorithm

Distance matrix

A → (1, 1)
B → (1, 2)
C → (3, 2)
D → (3, 3)
E → (6, 6)
F → (6, 7)
G → (7, 6)
H → (7, 7)
I → (7, 1)
J → (8, 2)
K → (9, 1)

C_1 \ C_2	A	B	C	D	E	F	G	H	I	J	K
A	0	1.0	2.2	2.8	7.0	7.8	7.8	8.48	6.0	7.0	8.0
B	1.0	0	2.0	2.2	6.4	7.0	7.0	7.8	6.0	7.0	8.06
C	2.2	2.0	0	1.0	5.0	4.98	5.6	6.4	4.1	5.0	6.08
D	2.8	2.2	1.0	0	4.2	5.0	5.0	5.6	4.5	5.09	6.3
E	7.0	6.4	5.0	4.2	0	1.0	1.0	1.414	5.09	4.47	5.83
F	7.8	7.0	4.9	5.0	1.0	0	1.414	1.0	6.08	5.38	6.70
G	7.8	7.0	5.0	5.0	1.0	1.414	0	1.0	5.0	4.12	5.38
H	8.48	7.8	6.4	5.6	1.414	1.0	1.0	0	6.0	5.09	6.32
I	6.0	6.0	4.1	4.5	5.09	6.08	5.0	6.0	0	1.414	2.0
J	7.0	7.0	5.0	5.09	4.47	5.38	4.12	5.09	1.414	0	1.414
K	8.0	8.06	6.08	6.3	5.83	6.7	5.38	6.32	2.0	1.414	0

Batchelor and Wilkins' Algorithm

$A \rightarrow (1,1)$
$B \rightarrow (1,2)$
$C \rightarrow (3,2)$
$D \rightarrow (3,3)$
$E \rightarrow (6,6)$
$F \rightarrow (6,7)$
$G \rightarrow (7,6)$
$H \rightarrow (7,6)$
$I \rightarrow (7,1)$
$J \rightarrow (8,2)$
$K \rightarrow (9,1)$

	A	B	C	D	E	F	G	H	I	J	K
A	0.0	1.0	2.2	2.8	7.0	7.8	7.8	8.48	6.0	7.0	8.0
B	1.0	0.0	2.0	2.2	6.4	7.0	7.0	7.8	6.0	7.0	8.06
C											
D											
E											
F											
G											
H											
I											
J											
K											

Batchelor and Wilkins' Algorithm

	C_1	C_2
	A	H
B	1.0	7.8
C	2.2	6.4
D	2.8	5.6
E	7.07	1.41
F	7.81	1.0
G	7.81	1.0
I	6.0	6.0
J	7.07	5.09
K	8.0	6.32

C_3

maximum among all minimum.

Batchelor and Wilkins' Algorithm

	C_1 A	C_2 H	C_3 K
B	1.0	7.8	8.06
C	2.2	6.4	6.08
C_4 D	2.8	5.6	6.32
E	7.07	1.41	5.83
F	7.81	1.0	6.70
G	7.81	1.0	5.38
I	6.0	6.0	2.0
J	7.07	5.09	1.41

Final clusters are $\{A, B, C, D\}, \{E, F, G, H\}, \{I, J, K\}$

Graph Based Clustering

- Graph can be represented as
$$G = \langle V, E \rangle$$
where V is set of nodes or vertices and E is set of Edges.
- There are many ways to represent a graph and one of them is similarity matrix
- For n number of nodes, similarity matix will be of size $n \times n$.
- The elements of similarity matrix will be either 0 or 1.
- $S(i, j) = 1$, if V_i and V_j are connected in some sense.

Similarity Matrix based clustering

Distance matrix

A → (1, 1)
B → (1, 2)
C → (3, 2)
D → (3, 3)
E → (6, 6)
F → (6, 7)
G → (7, 6)
H → (7, 7)
I → (7, 1)
J → (8, 2)
K → (9, 1)

	A	B	C	D	E	F	G	H	I	J	K
A	0	1.0	2.2	2.8	7.0	7.8	7.8	8.48	6.0	7.0	8.0
B	1.0	0	2.0	2.2	6.4	7.0	7.0	7.8	6.0	7.0	8.06
C	2.2	2.0	0	1.0	5.0	4.98	5.6	6.4	4.1	5.0	6.08
D	2.8	2.2	1.0	0	4.2	5.0	5.0	5.6	4.5	5.09	6.3
E	7.0	6.4	5.0	4.2	0	1.0	1.0	1.414	5.09	4.47	5.83
F	7.8	7.0	4.9	5.0	1.0	0	1.414	1.0	6.08	5.38	6.70
G	7.8	7.0	5.0	5.0	1.0	1.414	0	1.0	5.0	4.12	5.38
H	8.48	7.8	6.4	5.6	1.414	1.0	1.0	0	6.0	5.09	6.32
I	6.0	6.0	4.1	4.5	5.09	6.08	5.0	6.0	0	1.414	2.0
J	7.0	7.0	5.0	5.09	4.47	5.38	4.12	5.09	1.414	0	1.414
K	8.0	8.06	6.08	6.3	5.83	6.7	5.38	6.32	2.0	1.414	0

Similarity Matrix based clustering

(Similarity matrix) (Adjacency matrix) $d(x_i, x_j) \leq \theta$ $S_{ij} = 1$

	A	B	C	D	E	F	G	H	I	J	K
A	1	1									
B	1	1	1								
C		1	1	1							
D			1	1							
E					1	1	1	1			
F					1	1	1	1			
G					1	1	1	1			
H					1	1	1	1			
I									1	1	1
J									1	1	1
K									1	1	1

{B A C D}

{E F G H}

{I J K}

Minimal Spanning Tree based clustering

- Spanning tree is the tree representation of graph which contains all nodes present in the graph.
- Difference between tree and graph $\rightarrow$ Graph can have cycle but tree cannot have cycle.
- Weighted Graph have weighted edge which can be represented as

$$G = \langle V, E, W \rangle$$

$W \rightarrow$ weight or cost

Minimal Spanning Tree based clustering

- Many ways to represent a graph
  - Adjacency matrix
  - Edge list

$$A - B \rightarrow w_1$$
$$B - C \rightarrow w_4$$
$$A - E \rightarrow w_2$$
$$E - D \rightarrow w_5$$
$$D - C \rightarrow w_6$$
$$A - C \rightarrow w_7$$
$$B - D \rightarrow w_3$$

Minimal Spanning Tree based clustering

- All connected graph
- Weight is the distance (various distance measure) between two feature points.
- Spanning tree is the subset of the complete connected weighted graph.
- The minimal spanning tree is a spanning tree having minimum sum of the weights of edges.

Minimal Spanning Tree based clustering

A (1, 1)
B (1, 2)
C (3, 2)
D (3, 3)
E (6, 6)
F (6, 7)
G (7, 6)
H (7, 7)

	A	B	C	D	E	F	G	H
A	0	1.0	2.23	2.82	7.07	7.81	7.81	8.48
B	1.0	0	2.0	2.23	6.4	7.07	7.07	7.81
C	2.23	2.0	0	1.0	5.0	4.89	5.65	6.40
D	2.82	2.23	1.0	0	4.24	5.0	5.0	5.65
E	7.07	6.40	5.0	4.25	0	1.0	1.0	1.41
F	7.81	7.07	4.89	5.0	1.0	0	1.41	1.0
G	7.81	7.07	5.65	5.0	1.0	1.41	0	1.0
H	8.48	7.81	6.40	5.65	1.41	1.0	1.0	0

Ordered Edge List

A - B	1.0
C - D	1.0
E - F	1.0
E - G	1.0
F - H	1.0
G - H	1.0
E - H	1.41
F - G	1.41
B - C	2.0
A - C	2.23

B - D	2.23
A - D	2.82
D - E	4.24
C - F	4.89
C - E	5.0
D - F	5.0
D - G	5.0
C - G	5.65
D - H	5.65
B - E	6.40

C - H	6.40
A - E	7.07
B - F	7.07
B - G	7.07
A - F	7.81
A - G	7.81
B - H	7.81
A - H	8.48

- From this ordered edge list find the minimal spanning tree.

Minimal Spanning Tree based clustering

- There exist only one path between each pair of nodes in the tree.
- Minimal spanning tree is not unique.
- If root node is same than minimal spanning tree will be unique.

K-means clustering

Demonstration of the standard algorithm

1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).

2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.

3. The centroid of each of the *k* clusters becomes the new mean.

4. Steps 2 and 3 are repeated until convergence has been reached.

References

[1] John G Daugman. "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters". In: *JOSA A* 2.7 (1985), pp. 1160–1169.

[2] Cosmin Grigorescu, Nicolai Petkov, and Michel A Westenberg. "Contour detection based on nonclassical receptive field inhibition". In: *IEEE Transactions on image processing* 12.7 (2003), pp. 729–739.

[3] Simona E Grigorescu, Nicolai Petkov, and Peter Kruizinga. "Comparison of texture features based on Gabor filters". In: *IEEE Transactions on Image processing* 11.10 (2002), pp. 1160–1167.

[4] Anil K Jain and Farshid Farrokhnia. "Unsupervised texture segmentation using Gabor filters". In: *Pattern recognition* 24.12 (1991), pp. 1167–1186.

References

[5] Peter Kruizinga, Nicolai Petkov, and Simona E Grigorescu. "Comparison of texture features based on Gabor filters". In: *Image analysis and processing, 1999. Proceedings. International conference on*. IEEE. 1999, pp. 142–147.

[6] Nicolai Petkov and Michel A Westenberg. "Suppression of contour perception by band-limited noise and its relation to nonclassical receptive field inhibition". In: *Biological cybernetics* 88.3 (2003), pp. 236–246.

[7] Nikolai Petkov and Peter Kruizinga. "Computational models of visual neurons specialised in the detection of periodic and aperiodic oriented visual stimuli: bar and grating cells". In: *Biological cybernetics* 76.2 (1997), pp. 83–96.

[8] Nikolay Petkov. "Biologically motivated computationally intensive approaches to image pattern recognition". In: *Future Generation Computer Systems* 11.4-5 (1995), pp. 451–465.

Thank you!