Instruction of Clustering and Frequent Itemsets

Instruction 7

IDS

Chair of Process and Data Science

RWTH AACHEN UNIVERSITY

K-means

	Math	Physics
1	2	20
2	3	4
3	7	3
4	4	7
5	6	2
6	6	4
7	3	8
8	7	4
9	20	19

The following dataset shows the scores of two courses for nine students. We implement both k-means and k-medoids algorithms on this dataset and compare the results with each other.

Chair of Process and Data Science

K-means

Steps of K-means algorithm:

(1) Randomly choose k examples from the dataset as initial centroids.

(2) All the data points that are most similar to a centroid will create a cluster.

(3) Now, we have new clusters which need centers. The new value of the centroid is going to be the mean of all the examples in a cluster.

(4) We'll keep repeating steps 2 and 3 until the centroids stop moving.

K-means

(1) Points 2 and 8 are initial centroids. (3,4) and (7,4)

	Math	Physics
1	2	20
2	3	4
3	7	3
4	4	7
5	6	2
6	6	4
7	3	8
8	7	4
9	8	5
10	20	19

Chair of Process and Data Science

K-means

(2) All the data points that are most similar to a centroid will create a cluster. (Use Euclidean distance)

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2}$$

	Math	Physics	Distance from C1	Distance from C2
1	2	20	16	16.76
2	3	4	0	
3	7	3	4.12	1
4	4	7	3.16	4.24
5	6	2	3.6	2.23
6	6	4	3	1
7	3	8	4	5.65
8	7	4		0
9	8	5	5.09	1.41
10	20	19	22	19.84

K-means

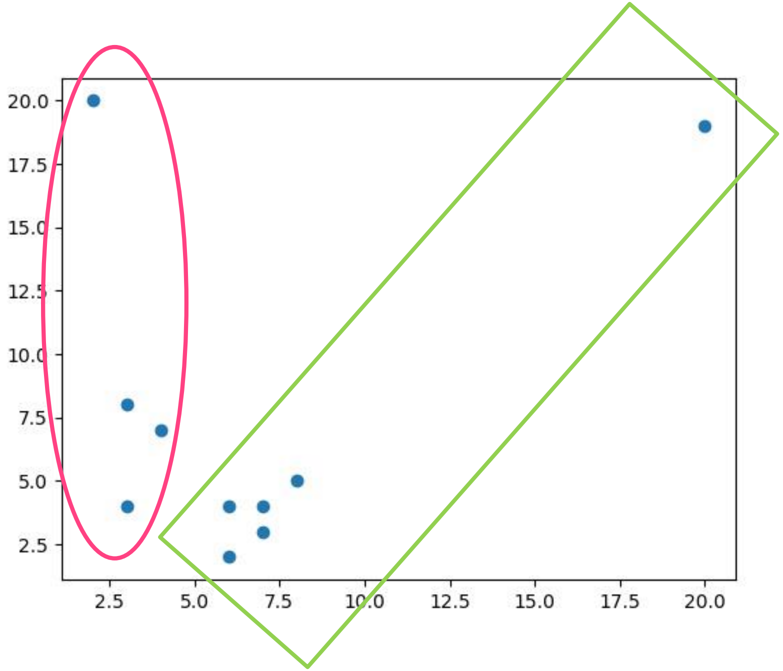
(3) Now, we have new clusters, which need centers. The new value of a centroid is going to be the mean of all the examples in a cluster.

	Math	Physics	Distance from C1	Distance from C2
1	2	20	16	16.76
2	3	4	0	
3	7	3	4.12	1
4	4	7	3.16	4.24
5	6	2	3.6	2.23
6	6	4	3	1
7	3	8	4	5.65
8	7	4		0
9	8	5	5.09	1.41
10	20	19	22	19.84

K-means

	Math	Physics	Distance from C1	Distance from C2
1	2	20	16	16.76
2	3	4	0	
3	7	3	[illegible]	[illegible]
4	4	7	[illegible]	[illegible]
5	6	2	[illegible]	[illegible]
6	6	4	[illegible]	[illegible]
7	3	8	4	5.65
8	7	4		0
9	8	5	5.09	1.41
10	20	19	22	19.84

New centers:
(3, 9.75)
(9, 6.16)



Chair of Process and Data Science

K-means

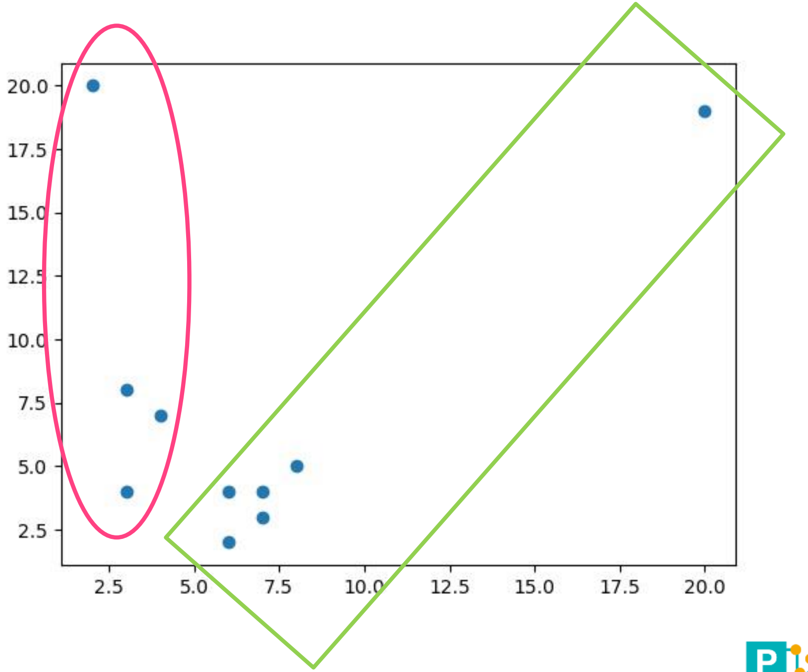
(2) All the data points that are most similar to a centroid will create a cluster.(Use Euclidean distance)

	Math	Physics	Distance from C1	Distance from C2
1	2	20	10.29	15.5
2	3	4	5.75	6.37
3	7	3	7.84	3.73
4	4	7	2.92	5.06
5	6	2	8.31	5.12
6	6	4	6.48	3.69
7	3	8	1.75	6.27
8	7	4	7	2.94
9	8	5	6.89	1.52
10	20	19	20.11	17.15

K-means

(4) We'll keep repeating step 2 and 3 until the centroids stop moving.

No change in clusters occurred!
We have final clusters.



Chair of Process and Data Science

K-means

Some weaknesses of k-means algorithm

- Number of clusters needs to be decided beforehand.
- It is sensitive to outliers.
- It can only discover spherical clusters (compare to density-based methods).

For practice, please choose two other centroids and repeat the algorithm. Compare the results with each other.

K-medoids

(1) Choose randomly two medoids.

	Math	Physics
1	2	20
2	3	4
3	7	3
4	4	7
5	6	2
6	6	4
7	3	8
8	7	4
9	8	5
10	20	19

K-medoids

(2) Assign each object to the closest representative object. Use Manhattan metric.

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}|$$

	Math	Physics	Distance from C1	Distance from C2
1	2	20	17	21
2	3	4	0	
3	7	3	5	1
4	4	7	4	6
5	6	2	5	3
6	6	4	3	1
7	3	8	4	8
8	7	4		0
9	8	5	6	2
10	20	19	32	28

K-medoids

(2) Assign each object to the closest representative object. Use Manhattan metric.

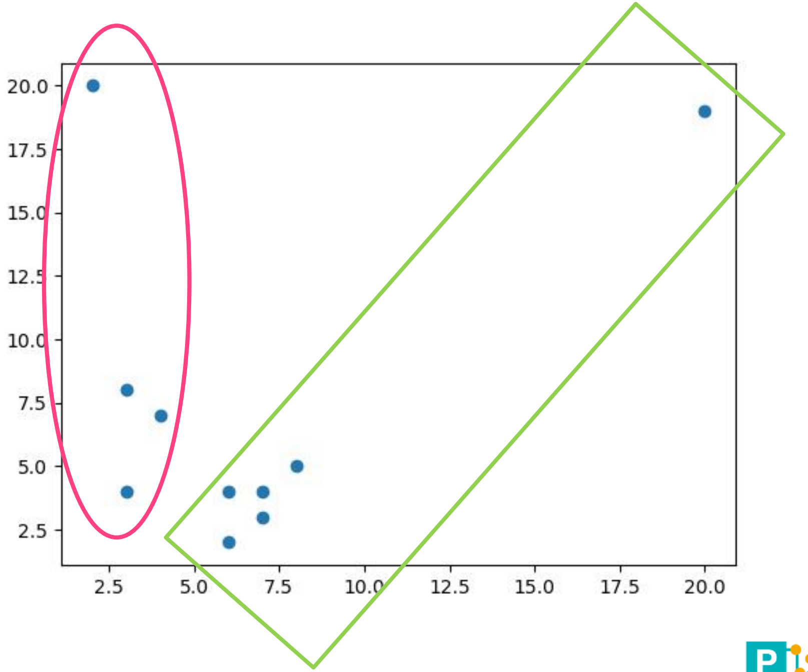
$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}|$$

	Math	Physics	Distance from C1	Distance from C2
1	2	20	17	21
2	3	4	0	
3	7	3	5	1
4	4	7	4	6
5	6	2	5	3
6	6	4	3	1
7	3	8	4	8
8	7	4		0
9	8	5	6	2
10	20	19	32	28

Chair of Process and Data Science

K-medoids

(2) Assign each object to the closest representative object. Use Manhattan metric.



K-medoids

Calculate the cost: The dissimilarity of each non-medoid point with the medoids is calculated:

cost: 17+4+4+1+3+1+2+28=60

K-medoids

(3) For each representative object, randomly select a non representative object O.

- **Choose a random object O1 (2,20). In this step notice that you do not the same experiment twice.**
- **Swap O8 and O1.**
- **Calculate the cost again.**

K-medoids

(4) Assign each object to the closest representative object. Use Manhattan metric.

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}|$$

	Math	Physics	Distance from C1	Distance from C2
1	2	20		0
2	3	4	0	
3	7	3	5	22
4	4	7	4	15
5	6	2	5	22
6	6	4	3	20
7	3	8	4	13
8	7	4	4	21
9	8	5	6	21
10	20	19	32	19

K-medoids

(4) Assign each object to the closest representative object. Use Manhattan metric.

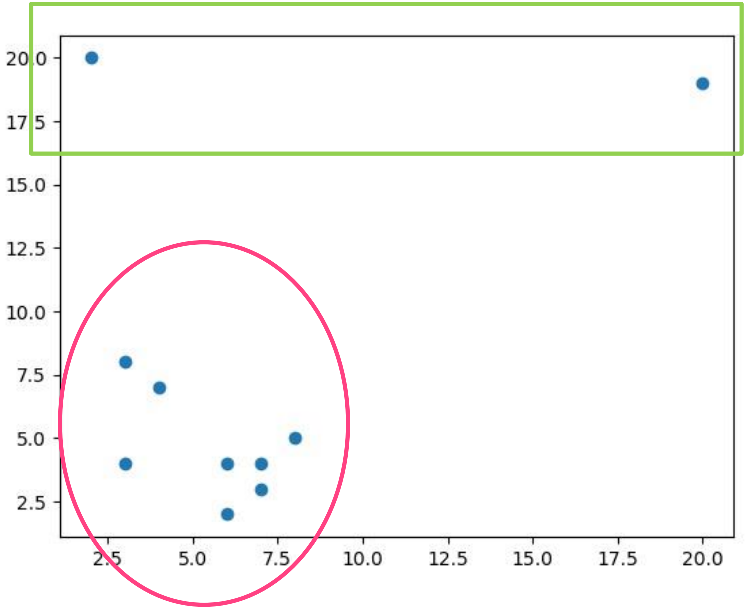
$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{ip} - x_{jp}|$$

Cost:
5+4+5+3+4+4+6+19=50

	Math	Physics	Distance from C1	Distance from C2
1	2	20		0
2	3	4	0	
3	7	3	5	22
4	4	7	4	15
5	6	2	5	22
6	6	4	3	20
7	3	8	4	13
8	7	4	4	21
9	8	5	6	21
10	20	19	32	19

K-medoids

(4) Assign each object to the closest representative object. Use Manhattan metric.



Chair of Process and Data Science

K-medoids

If new cost is less than previous cost replace the representative object with o random.

50<60
It is good to replace O8 and O3.

(6) We try other non-medoids points to get minimum distance…

(7) Back to step 1, until no change.

Comparison of k-means and k-medoids

- **K-medoids is more robust to noise and outliers but:**
- **The complexity of each iteration is high: O(k(n-k)^2)**
  (k: number of representative objects, n: total number of objects)

Chair of Process and Data Science

Frequent Itemsets

Basic ideas of Apriori algorithm

- Apriori rule: All the non-empty sub-itemsets of frequent itemsets must be frequent.

Apriori Algorithm

TID	Items
1	sugar, fruit, water
2	bread, fruit, juice
3	sugar, bread, fruit, juice
4	bread, juice
5	sugar, fruit, juice

C1

Itemset	Count
sugar	3
bread	3
juice	4
fruit	4
water	1

Due to the min-support-count= 2, we remove water from the table.

Chair of Process and Data Science

Apriori Algorithm

L1

Itemset	Count
sugar	3
bread	3
juice	4
fruit	4

C2

Itemset	Count
{sugar, bread}	1
{sugar, juice}	2
{sugar, fruit}	3
{bread, juice}	3
{bread, fruit}	2
{juice, fruit}	3

Due to the min-support-count= 2, we remove {sugar, bread}.

Chair of Process and Data Science

Apriori Algorithm

L2

Itemset	Count
{sugar, juice}	2
{sugar, fruit}	3
{bread, juice}	3
{bread, fruit}	2
{juice, fruit}	3

C3

Itemset	In L2
{sugar, juice, fruit} {sugar, juice},{juice, fruit},{sugar, fruit}	Yes
{sugar, juice, bread} {sugar, juice},{sugar, bread},{juice, bread}	No
{sugar, fruit, bread} {sugar, fruit},{sugar, bread},{fruit, bread}	No
{bread, fruit, juice} {bread, fruit},{fruit, juice},{bread, juice}	Yes

If an itemset is frequent, each subset of that should be frequent

Apriori Algorithm

C3

Itemset	Support
{sugar, juice, fruit}	2
{bread, fruit, juice}	2

Min-support-count= 2 →

L3

Itemset	Support
{sugar, juice, fruit}	2
{bread, fruit, juice}	2

For making L4, look at the first dataset.

Apriori Algorithm

C4

Itemset	In L3
{sugar, juice, fruit, bread} {sugar, juice, fruit},{sugar, juice, bread},{fruit, juice, bread}	No

✓ **The Apriori algorithm takes the advantage of the fact that any subset of a frequent itemset should also be frequent.**

Chair of Process and Data Science