worksheet_09

May 1, 2024

1 Worksheet 09

Name: Honghao Zhao

UID: U44266035

1.0.1 Topics

- Clustering Review
- Clustering Aggregation

1.0.2 Clustering Aggregation

Point	C	P
A	0	a
B	0	b
C	2	b
D	1	c
E	1	d

a) Fill in the following table where for each pair of points determine whether C and P agree or disagree on how to cluster that pair.

Pair	Disagreement
A B	disagree
A C	disagree
A D	disagree
A E	disagree
B C	disagree
B D	disagree
B E	disagree
C D	disagree
C E	disagree
D E	disagree

As datasets become very large, this process can become computationally challenging.

b) Given N points, what is the formula for the number of unique pairs of points one can create?

```
[2]: from IPython.display import display, Math
     display(Math(r'\binom{N}{2} = \frac{N \times (N - 1)}{2}'))
```

$$\binom{N}{2} = \frac{N \times (N-1)}{2}$$

Assume that clustering C clusters all points in the same cluster and clustering P clusters points as such:

Point	P
A	0
B	0
C	0
D	1
E	1
F	2
G	2
H	2
I	2

c) What is the maximum number of disagreements there could be for a dataset of this size? (use the formula from b)?

8*9/2 = 36

d) If we look at cluster 0. There are (3 x 2) / 2 = 3 pairs that agree with C (since all points in C are in the same cluster). For each cluster, determine how many agreements there are. How many total agreements are there? How many disagreements does that mean there are between C and P?

total agreements = C(3,2) + C(2,2) + C(4,2) = 3+1+6 = 10 disagreements = 36 - 10 = 26

e) Assuming that filtering the dataset by cluster number is a computationally easy operation, describe an algorithm inspired by the above process that can efficiently compute disagreement distances on large datasets.

1. First, segregate the dataset into subsets according to their cluster labels assigned by either of the clustering results. This allows for a quick access to the points that belong to the same cluster.

2. Iterate over each unique pair of cluster labels, for instance, cluster i from clustering C and cluster j from clustering P where i j. For each such pair, compute all possible point pairs where one point is from cluster i and the other is from cluster j.

3. For each pair of clusters, count the number of point pairs formed. This step can be executed in parallel since the calculations for different cluster pairs are independent of each other.

4. Sum up all these disagreements to get the total number of disagreement pairs. This is the total disagreement distance for the dataset.