Worksheet 09

Name: Minfeng Qian UID: U10282863

Topics

- Clustering Review
- Clustering Aggregation

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	No
A C	Yes
A D	No
A E	No
B C	Yes
B D	No
B E	No
C D	Yes
C E	Yes
D E	Yes

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?

The expression for the distinct number of points is calculated using the combinations of pairs, simplifying to C(N, 2) = (N * (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)?

Substituting the values into the formula: (9 * 8) / 2 = 36 possible conflicts.

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?

In cluster 0, there are 3 points (A, B, C). Since all points are placed in the same cluster according to clustering C, there are no disagreements among these points. Each point in cluster 0 is in agreement with every other point regarding their cluster membership in both C and P. Therefore, the number of agreements can be calculated using the combination formula: (3 * 2 / 2) = 3 agreements. Following the same method for the other clusters, we find 1 pair and 6 pairs respectively. Thus, the total agreements amount to 1 + 3 + 6 = 10. With 36 potential disagreements and 10 agreements, there are 26 total disagreements.

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.

Algorithm: Efficient Disagreement Distance Calculation

Step 1: Preprocessing

- **Input:** Dataset (D) with points assigned to clusters according to two different clustering schemes (C) and (P).
- **Operation:** Pre-compute and store the cluster memberships for each point under both clustering schemes. This involves creating two lists or arrays, (C[i]) and (P[i]), where (i) represents the index of a point in (D), and the values represent the cluster number to which point (i) is assigned in (C) and (P) respectively.

Step 2: Filtering by Cluster

- **Operation:** For each cluster (k) in (C), retrieve all points that belong to (k). Repeat this operation for each cluster in (P). This filtering can be efficiently done using hash tables or dictionaries where the keys are the cluster numbers and the values are lists of points belonging to those clusters.

Step 3: Calculate Agreements

- **Operation:** For each cluster (k) in (C):
  - For each cluster (j) in (P):
    - Intersect the sets of points in (C[k]) and (P[j]). The size of this intersection gives the number of pairwise agreements between points that both schemes agree are in the same cluster.
  - The total number of agreements within these intersecting subsets can be calculated using the combination formula (C(n, 2)), where (n) is the number of points in the intersection.

Step 4: Compute Disagreements

- **Calculation:** Knowing the total number of potential pairwise interactions in the dataset (using (C(N, 2)) where (N) is the total number of points in (D)) and the number of agreements from Step 3:
  - Total disagreements = Total potential pairwise combinations - Total agreements.

Step 5: Optimization Considerations

- **Batch Processing:** For very large datasets, process data in batches to minimize memory usage.
- **Parallel Processing:** Use multi-threading or distributed computing to handle different clusters or pairs of clusters in parallel, especially when intersecting and counting pairs.

- **Data Structures:** Efficient data structures like hash maps (for cluster memberships) and trees (for sorting and quickly finding intersections) can significantly speed up the process.

Step 6: Output

- **Return:** The total number of disagreements between t suitable for large datasets.