COMP9318 Project Report

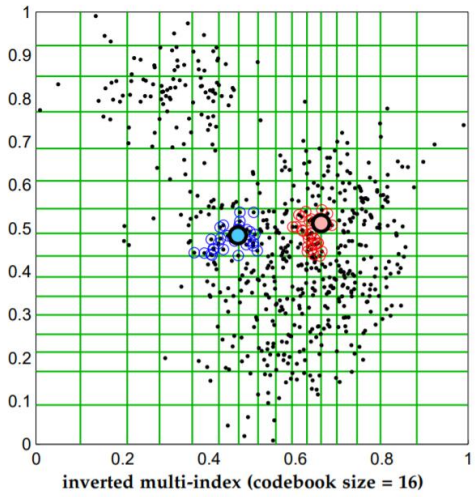
z5187292 Ran Bai
z5148417 Le Wang

Question 1: Implementation details of part 1, especially what changes you made to accomodate L1 distance.

Answer:

In the clustering algorithm, the median of all points in the current cluster is used when calculating the new centroid instead of means. Firstly, split the codebooks and codes into P parts. Then, I do the following two steps in a loop, put the points of data into the cluster represented by centroid, and recalculate the centroid with the points in cluster. Finally, we got the new codebooks as a result, and use new codebooks to code the given data.

As the image below, the vectors of P dimensions in codebooks (the result of part 1)represent points on P coordinate axes, respectively. The codes are N clusters where the query point is located.



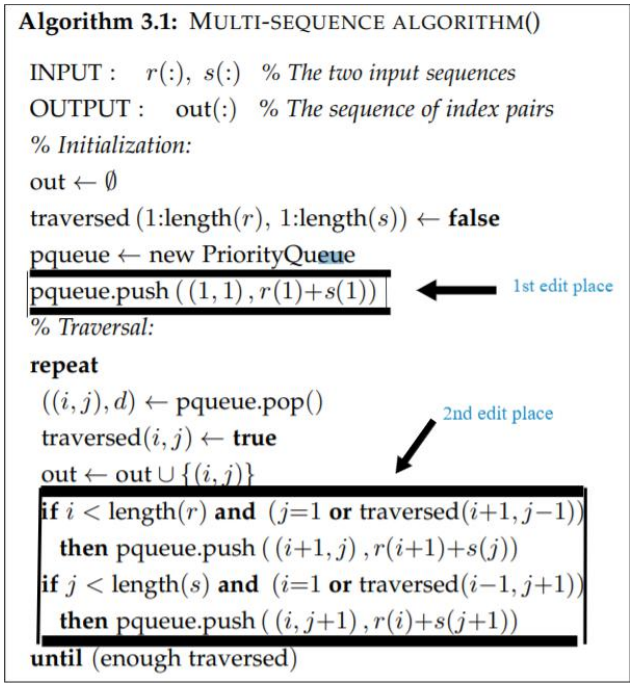
Question 2: Implementation details of part 2, including the details on how you extended the algorithm 3.1 to a more general case with P>2, and how you efficiently retrieve the candidates.

Answer:

In this part, I start searching from the cluster closest to the query point. If the cluster contains points in codes, add candidate.

I extend the multi-sequence algorithm when P>2 as below

Algorithm 3.2: MULTI-SEQUENCE ALGORITHM()



When P > 2, The first edit place should be
Pqueue.push((0, 0, 0... ,0))
pqueue.push((0, *, *... *), (*, 0, *,.... *), ...(*, *, ... 0))
where * represent all possible value in corresponding dimension of codes.
And the second edit place should replace by
Pqueue.push(i + 1, *,....., *)
Pqueue.push(*, j + 1, *..... *)
.....
Pqueue.push(*, *, *..... z +1)
where * represent all possible value in corresponding dimension of codes. It's can help us efficiently retrieve the candidates because I only search the point related to points in codes.