

COMP9318 Project Report

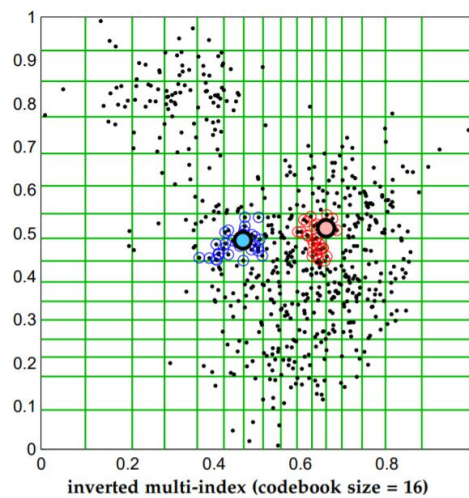
z5187292 Ran Bai
z5148417 Le Wang

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()

Algorithm 3.1: MULTI-SEQUENCE ALGORITHM()

```

INPUT :    $r(\cdot), s(\cdot)$  % The two input sequences
OUTPUT :   out(:) % The sequence of index pairs
% Initialization:
out  $\leftarrow \emptyset$ 
traversed (1:length(r), 1:length(s))  $\leftarrow$  false
pqueue  $\leftarrow$  new PriorityQueue
pqueue.push ( (1, 1),  $r(1)+s(1)$ )  ← 1st edit place
% Traversal:
repeat
  ((i, j), d)  $\leftarrow$  pqueue.pop()
  traversed(i, j)  $\leftarrow$  true
  out  $\leftarrow$  out  $\cup \{(i, j)\}$ 
  if  $i < \text{length}(r)$  and ( $j=1$  or traversed( $i+1, j-1$ ))
    then pqueue.push ( ( $i+1, j$ ),  $r(i+1)+s(j)$ )
  if  $j < \text{length}(s)$  and ( $i=1$  or traversed( $i-1, j+1$ ))
    then pqueue.push ( ( $i, j+1$ ),  $r(i)+s(j+1)$ )
until (enough traversed)

```

2nd edit place

When $P > 2$, The first edit place should be

$\text{Pqueue.push}((0, 0, 0 \dots, 0))$

$\text{pqueue.push}((0, *, * \dots *), (*, 0, *, \dots *), \dots, (*, *, \dots, 0))$

where * represent all possible value in corresponding dimension of codes.

And the second edit place should replace by

$\text{Pqueue.push}(i + 1, *, \dots, *)$

$\text{Pqueue.push}(*, j + 1, * \dots *)$

.....

$\text{Pqueue.push}(*, *, * \dots z + 1)$

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.