1 Problem Definition

In this assignment, you are required to implement and benchmark nearest neighbor similarity queries using the two-step and multi-step similarity search algorithms. The algorithms should take as input a query vector q and should compute the nearest neighbor p of q.

2 Design and Implementation

The algorithm has three steps. First, we do a Principle Component Analysis (PCA) on the dataset. Second, we construct a R-Tree with the dataset in the principle space. At last, we run the two step and multi-step nearest neighbor search.

2.1 Principle Component Analysis (PCA)

For a dataset array X_{nxd}, we first center X and get X'. Then, we run a SVD on X', so that $X' = USV^T$. We get the principle component $G = U_{n*k}S_{k*k}$. We then insert G into the R-Tree. For a query q, we get its corresponding vector q' by $q' = qV_{n*k}$.

2.2 Nearest Neighbor Search

To find the nearest neighbor for a query q', we conduct a two-step and a multi-step nearest neighbor search. For two-step search, we use the D we calculate in the reduced space and do a range search of $[q' - D, q' + D]$. After that, we get the final result by comparing all data in the range search.

For the multi-step search, we do incremental R-Tree queries until the distance in reduced space is larger than the distance in the original space.

2.3 Benchmark

Evaluation of the algorithm is focused of the time of constructing the R-Tree and the average query time. Queries were generated randomly and these queries are served to the nearest neighbor search system. The average query time for 1k queries is recorded. A simple linear scan algorithm is used for comparison. The result is showed in Table 1.

k	**Indexing**	**Two-Step**	**Multi-Step**	**Linear**
5	80.6s	439ms	639ms	350ms
10	125s	468ms	695ms	350ms
15	165s	512ms	726ms	350ms
20	167s	578ms	748ms	342ms
25	193s	627ms	287ms	350ms
28	206.8s	642ms	198ms	344ms

Table 1: Evaluation result.

Table 1 shows that linear search outperforms two-step and multi-step nearest neighbor search when k is less than 25. This is because the principle component of X get 99% information of the original X when k is larger than 25. For two-step nearest neighbor search, the distance D is not a good estimation of the reduced dimension, causing a bad performance.