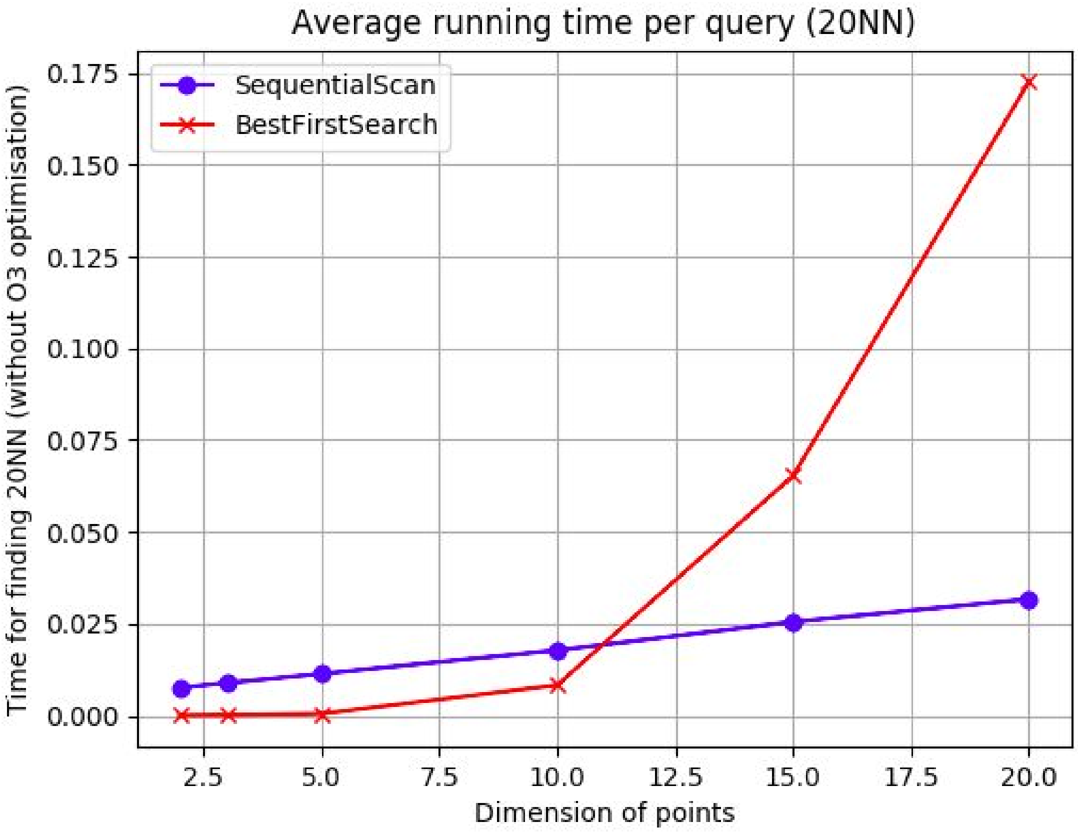
COL362 Assignment 3

Report

Utkarsh Singh, 2015ME10686

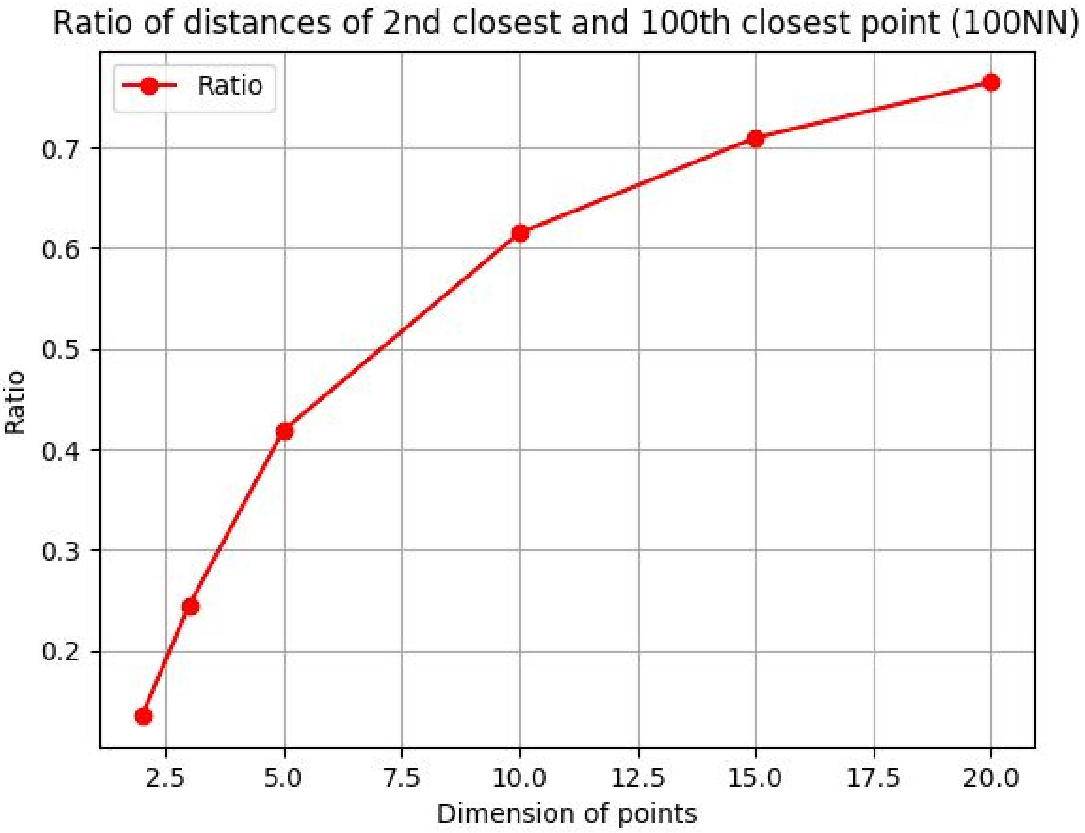
Part A:

For part A, we compare the performance of the Sequential Scan algorithm and Best First Search algorithm for finding 20NN query for 100 points on a dataset of 1,00,000 points. The average query time is plotted for dimensions 2, 3, 5, 10, 15 and 20.



Part B:

For part B, we compute the distances of the 2nd closest point and the 100th closest point. We do this for 100 query points and compute the ratio of the average distances (2nd closest point / 100th closest point).



Part C:

Explanation of Part A observations: We see that for lower dimensional data, we can use KD trees to efficiently compute 20NN query (or more generally, kNN queries) compared to a normal Sequential Scan algorithm. However, as the dimensionality increases, the performance of the Best First Search algorithm decreases comparatively. This is because of the **curse of dimensionality** observed in higher dimensional data.

In higher dimensions, the points are more sparse. Because of this sparsity, we are unable to prune most of the search space in our KD tree. This causes us to visit most of the possible search space, thereby increasing the running time. (Also, it’s generally observed that number of KD tree search can be exponential in dimension. Reference: https://courses.cs.washington.edu/courses/cse599c1/13wi/slides/lsh-hashkernels-annotated.pdf)

Explanation of Part B observations: We see that the ratio increases with the increase in dimension of the space. The ratio seems to grow exponentially. This observation too can be explained by the notion of **curse of dimensionality.**

Consider a dimension ‘d’. Since we obtain our points from a uniform distribution, we can assume that the density of points in a hypersphere is a constant, K. The volume of a hypersphere is given by

$$V_n(R) = \frac{\pi^{\frac{n}{2}}}{\Gamma\left(\frac{n}{2}+1\right)} R^n,$$

Let the distance of the 2nd closest point be R_2 and the 100th closest point be at distance R_{100}. Then, the number of points between the second closest point and the farthest point is (let them be M)

$$M = K * [V_n(R_{100}) - V_n(R_2)]$$

R_2 will always be a constant. The above equation can be rearranged to obtain

$$R_2/R_{100} \ \alpha \ \ 1/[f(n)]^{1/n}$$

Where $[f(n)]^{1/n}$ decreases exponentially, thus increasing the ratio.