Universidad Católica San Pablo

Advanced Data Structures - Final Project Results

- Renato Postigo
- Undergraduate Student - Computer Science Department, Universidad Católica San Pablo
- renato.postigo@ucsp.edu.pe

- July 9th, 2019

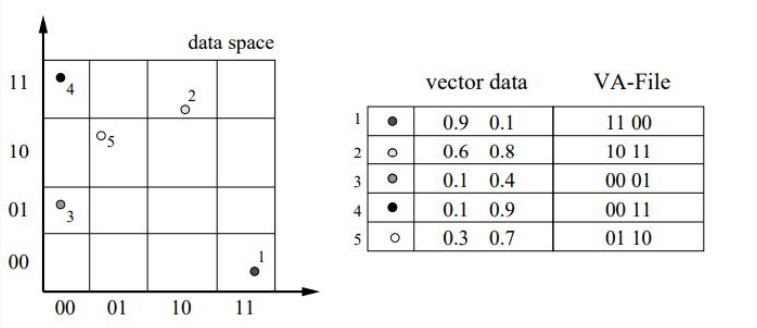
Proposal

Proposal

- Two data structures for storage of multi-dimensional data.
  - Geographical reference: QuadTree.
  - Whole data: VA-File.
- Objective: getting the k nearest neighbours.
- Metric to be used: Euclidean distance.

How does a VA-File work?

- We get an approximation for each data point (represented in bits).
- For each dimension j we get a number of bits to represent it. (b_j).
- Approximation: a_i.

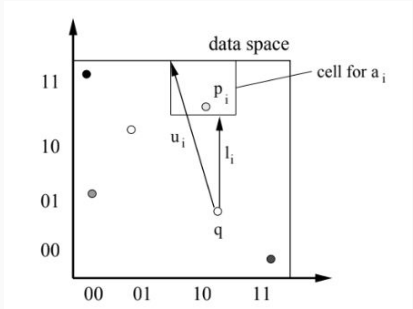


		vector data	VA-File
1	●	0.9 0.1	11 00
2	○	0.6 0.8	10 11
3	●	0.1 0.4	00 01
4	●	0.1 0.9	00 11
5	○	0.3 0.7	01 10

Graphic retrieved from [2].

How do we find the k-nearest neighbours?

- Each point *p* in the VA-File is in a cell of its own.
- Given a query point *q*, we can a find a lower bound (shortest distance to the cell) and an upper bound (longest distance to the cell).
- We use the lower bound as a filter so we don't have to go through all the points.



Graphic retrieved from [2].

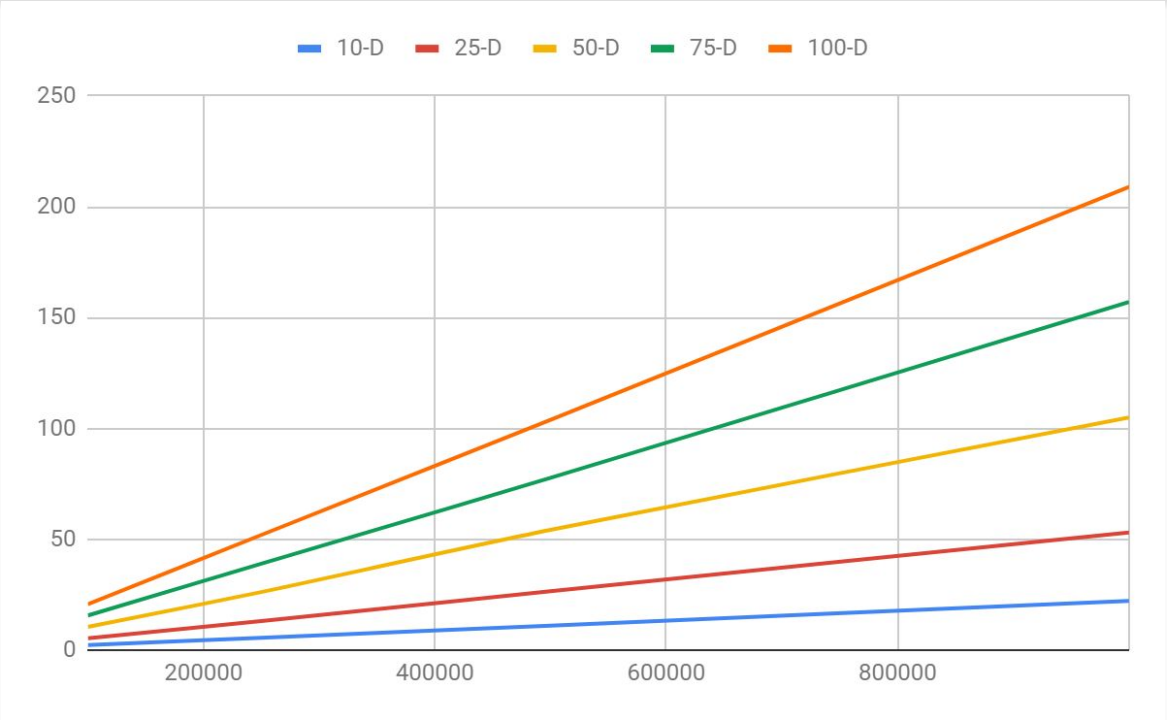
Results

Experiment Procedure

- Different datasets randomly generated using python.
  - Different dimensions: 10, 25, 50, 75 and 100.
  - Different total sizes: 100k, 250k, 500k and 1M.
- Time to build the VA-File structure.
- Time to find k-nearest neighbours with different k values.
  - k = 5, 10, 100, 500, 1000, 5000.
- Hardware: Lenovo Ideapad 510S
  - 4GB of RAM, Core i5 7200-U 2.5GHz

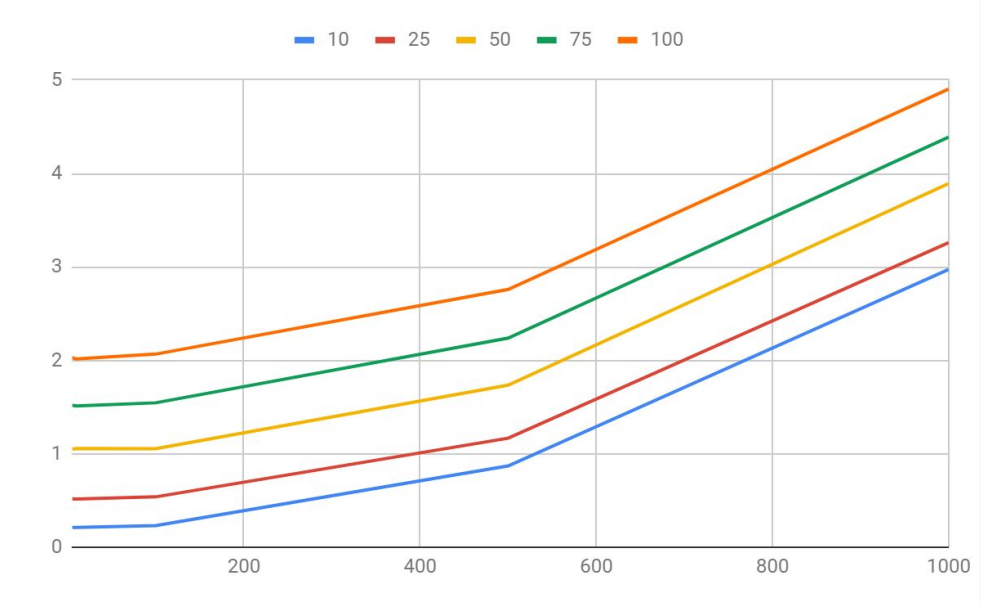
Time to build the VA-File

- Time in seconds.
- Linear time.

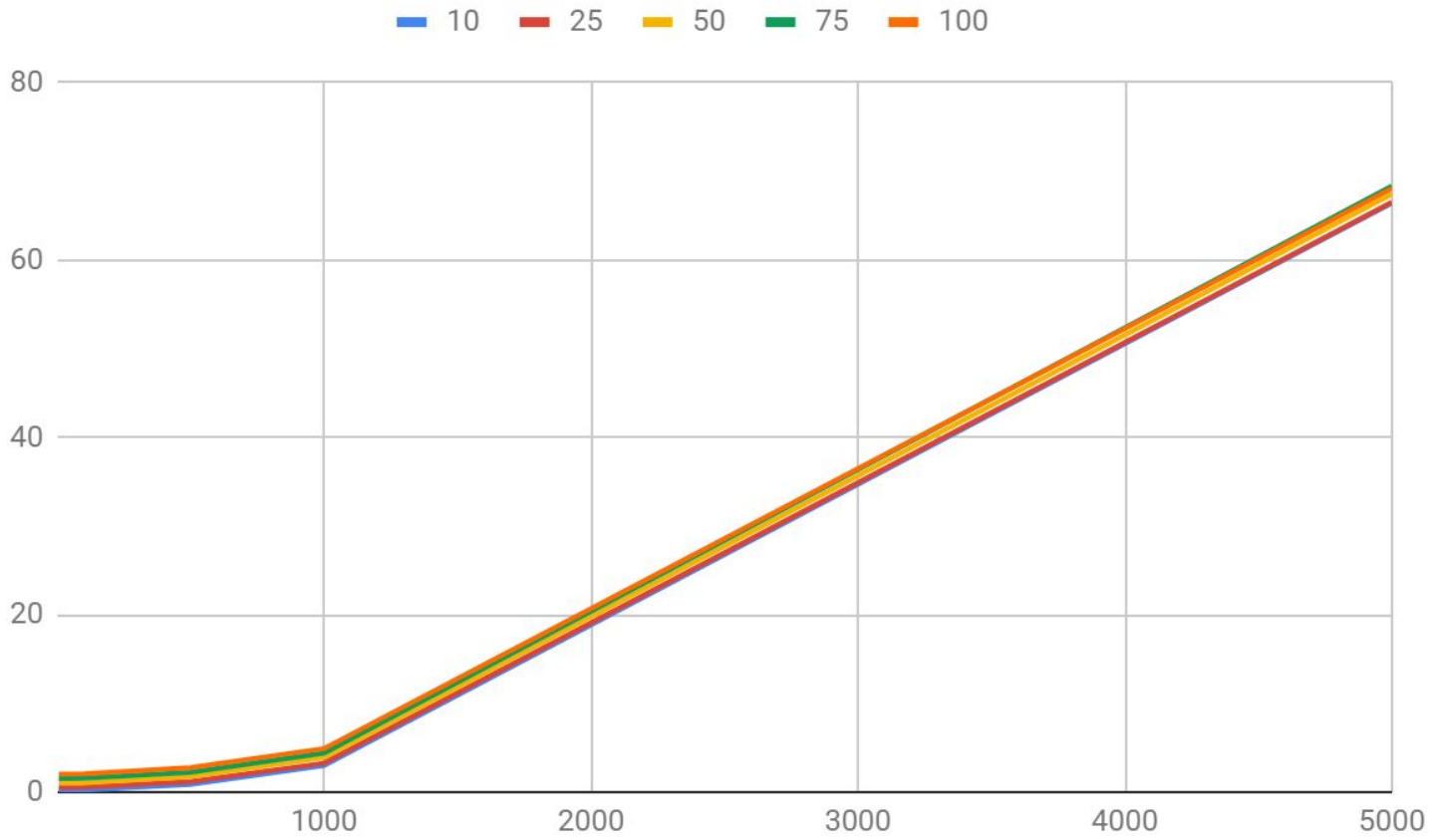


Time to find k-near neighbours at datasets of 100k points (k <= 1000)

- Similar behaviour.
- The graph is similar for all datasets, but something interesting happens when we try to find bigger k-nearest neighbours.

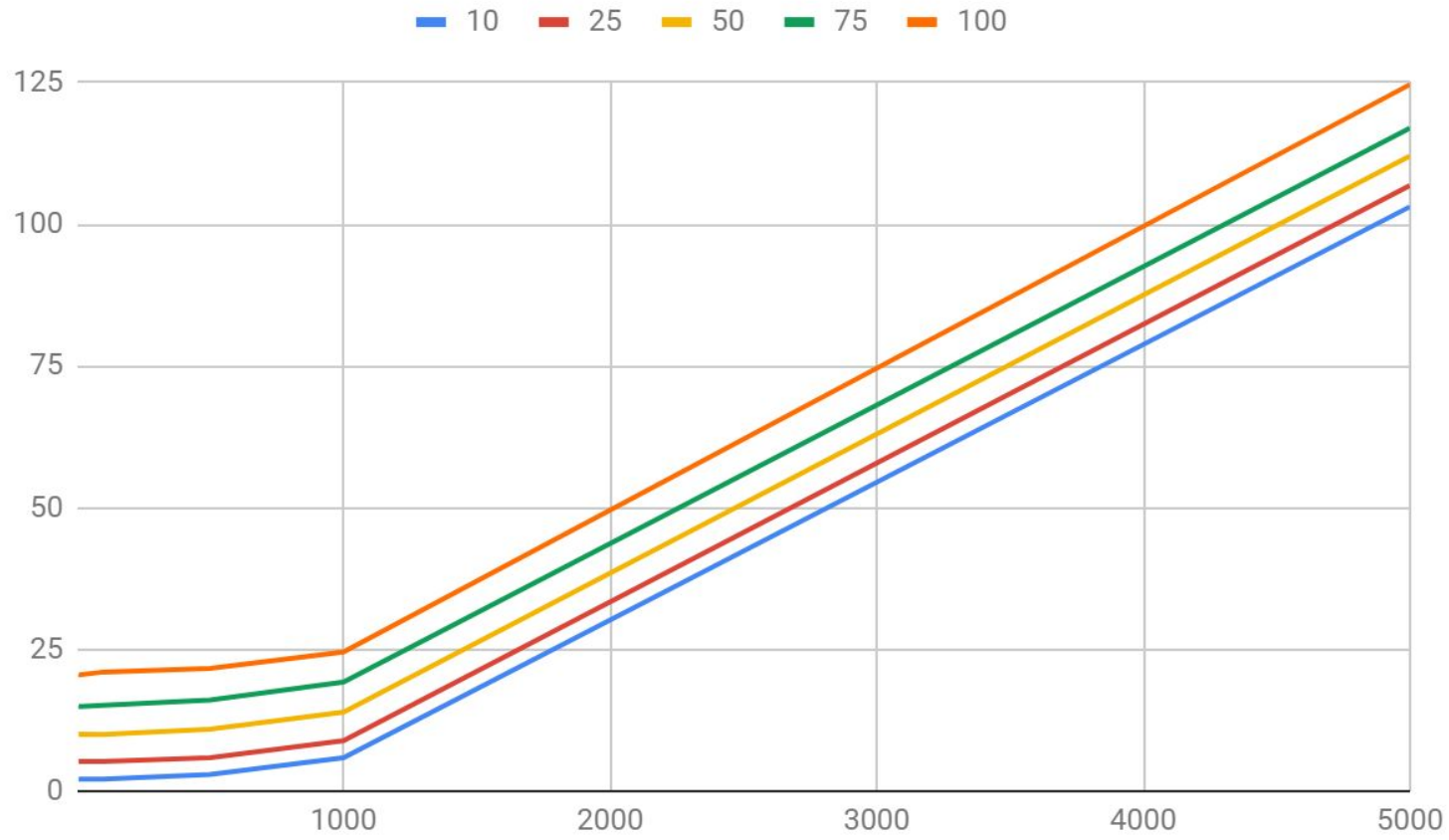


Times to find k-nearest neighbours - 100k data points



Same 100k data points. We see the time goes from 5 seconds to over 60, but it's very similar across all the different dimensions.

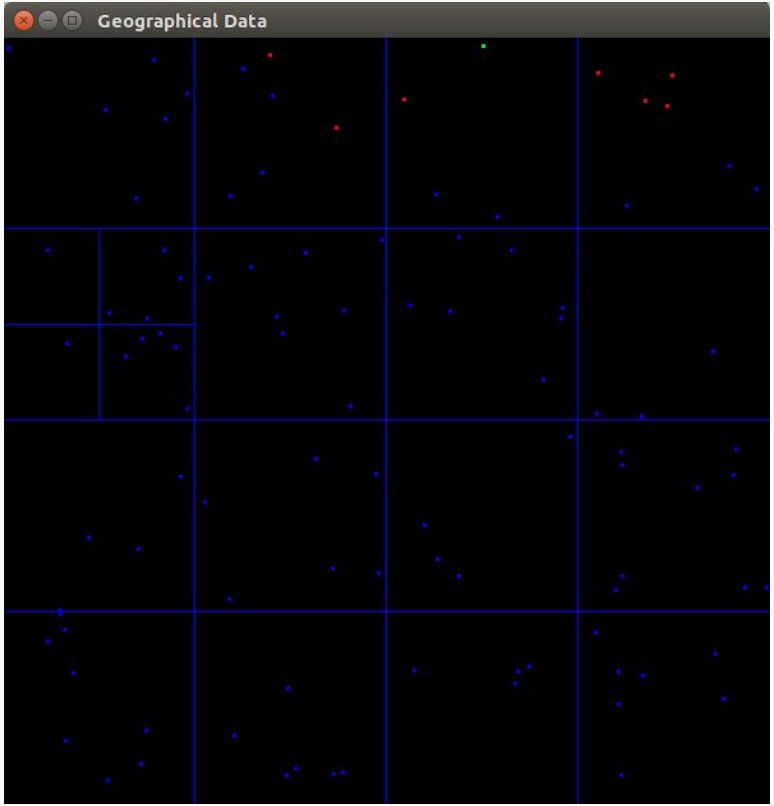
Times to find k-nearest neighbours - 1M data points.



The more data points, the more spread the times are, but the growth isn't too noticeable between dimensions.

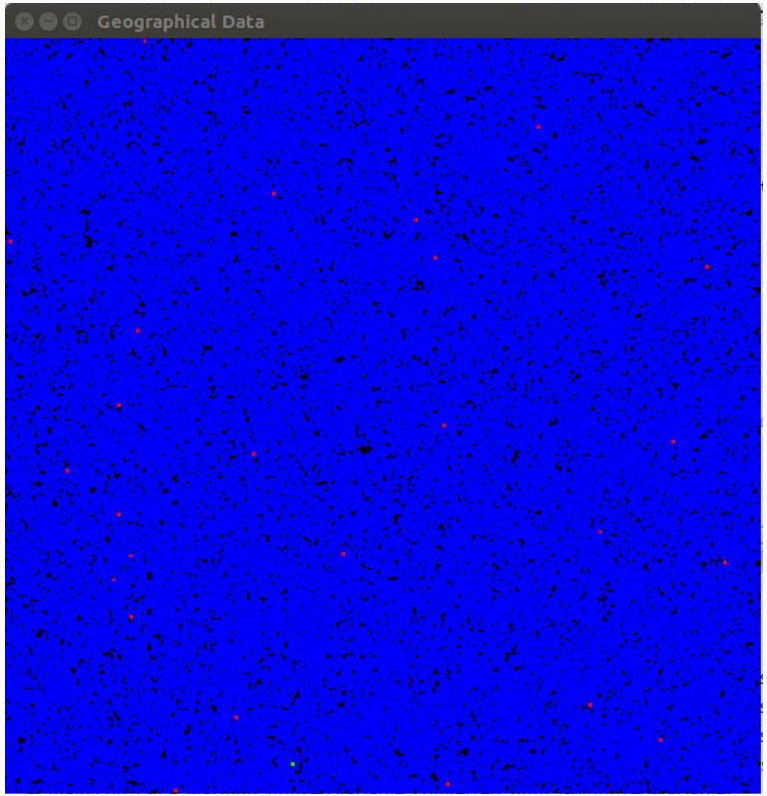
It still takes much more time when k is big compared to small k's.

The structure in action - Testing k-nn with really small dataset (100 points)

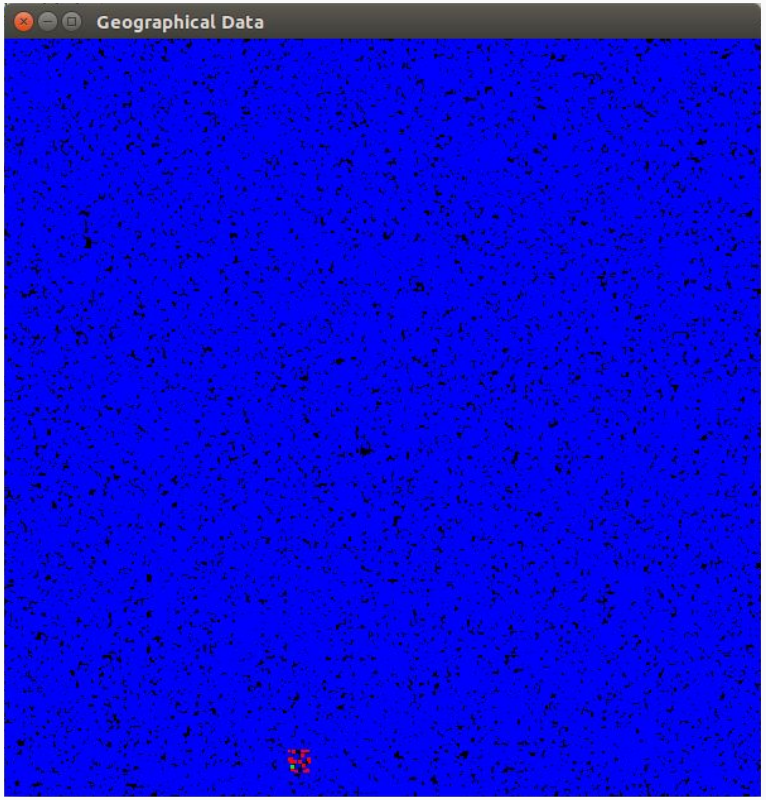


7-nn for 2-dimensional data.

The structure in action - Finding nearest neighbours in a quadtree quadrant (100k, 10 dimensions, 25-nn)



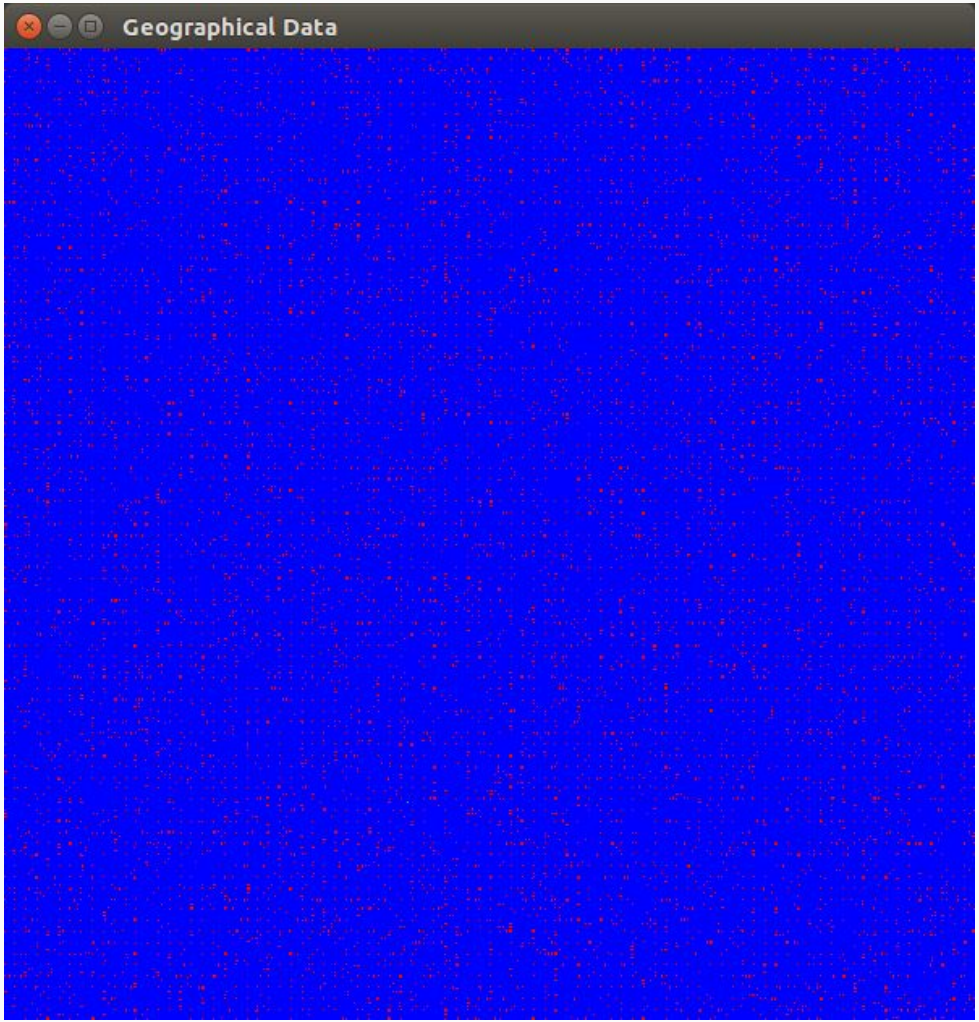
Searching at all the points.



Searching within quadtree quadrant.

Data visualization using quadtree

- Finding 40000 nearest neighbours of a certain point (red points).
- 1M points (put in blue).
- Each data point has 50 dimensions.
- Time it took: 5750 seconds (over hour and a half).
- Can barely see the red points.



Conclusions

Conclusions

- If we only want to do k-nn queries, VA-File it’s a really good and solid option as it does it quick and is easy to implement.
  - Starts to take more time as k increases. Huge k’s: Takes too much time.
  - Really easy to implement compared to other structures.
  - It does a good job fighting the dimensionality curse.
- VA-File not really useful for anything else.

References

- [1] R. A. Finkel and J. L. Bentley. Quad trees a data structure for retrieval on composite keys. Acta Informatica, 4(1):1–9, Mar 1974.
- [2] R. Weber and S. Blott. An approximation-based data structure for similarity search. 1997.
- My VA-File Implementation: github (header) github (body)

Universidad Católica San Pablo

Advanced Data Structures - Final Project Results

- Renato Postigo
- Undergraduate Student - Computer Science Department, Universidad Católica San Pablo
- renato.postigo@ucsp.edu.pe

- July 9th, 2019