# **K-Nearest-Neighbours: A comparison of KD-Trees and LSH Performance on Multidimensional Datasets**

# Team Information

\*Team information should list members with names, BU ID and SCC user names

Samir Farhat Dominguez - U41707119 - safarhat

# Abstract

\*Abstract should give a short summary of your work. E.g., This project aims to implement and compare VeB Trees and X-Fast trie in language C. The results validate the implementation with various input files, and comment on the performance analysis.

### Objectives

“EC504 Final Project: One of the standard problems in information retrieval is the nearest-neighbor problem. Given an object and a database, quickly retrieve the k-most similar objects to the query from the database. A naive implementation would require one to linearly scan through the entire database, which is typically too costly for online applications. Two common approaches that have been suggested for this problem are locality-sensitive hashing and k-d-trees. Both may be viewed as data structures / algorithms for finding an approximate nearest neighbor to a query. Other approaches such as R-trees exist. In this project, you will explore and implement these two techniques, and test them on a data set (for example, the 80 million tiny images dataset from MIT, where nearest neighbors can be computed directly using L2 distance over the computed gist features of these images; or a zip code locator based on latitude and longitude information). Compare the performance of these approaches in terms of speed and accuracy (i.e., how often do these methods retrieve one of the top few actual nearest neighbors).” \*Rephrase this

### Summary

\*Brief summary of Implementation (Language, platform, files, functions etc)

### Results

\*Brief summary of Results and what we can extrapolate/conclude from those results

# Instructions

\*Instructions must clearly list information on how to build the code and run with provided input files. It should also validate output. This is the most important part of the report, which will enable TAs to grade your project appropriately. Basically a README

# Sample Results

\*Sample results should show sample output for the code, along with discussion of the results

### KD-Trees

### Locality Sensitive Hashing

### Comparison

\*Don’t think graphs are necessary but just relevant print statements

# References

\*References include a list of sources your team used in developing the software. They can include web sites, or reference papers.

1. Torralba, Antonio, Rob Fergus, and William T. Freeman. "80 million tiny images: A large data set for nonparametric object and scene recognition." IEEE transactions on pattern analysis and machine intelligence 30.11 (2008): 1958-1970.
2. Abouelnaga, Yehya, et al. "Cifar-10: Knn-based ensemble of classifiers." 2016 International Conference on Computational Science and Computational Intelligence (CSCI). IEEE, 2016.
3. Moore, Andrew W. "An intoductory tutorial on kd-trees." (1991).
4. Bhatia, Nitin. "Survey of nearest neighbor techniques." arXiv preprint arXiv:1007.0085 (2010).
5. Ram, Parikshit, and Kaushik Sinha. "Revisiting kd-tree for nearest neighbor search." Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining. 2019.
6. Dasgupta, Anirban, Ravi Kumar, and Tamás Sarlós. "Fast locality-sensitive hashing." Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. 2011.
7. Slaney, Malcolm, and Michael Casey. "Locality-sensitive hashing for finding nearest neighbors." IEEE Signal processing magazine 25.2 (2008): 128.