Research Statement

Bao Ngo[*]
Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada
ngot1@myumanitoba.ca

Jared Rost[*]
Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada
rostj@myumanitoba.ca

1 Project Motivation

The past decade has seen significant advances in Machine Learning (ML) and a surge in ML applications to various domains, such as Natural Language Processing (Bubeck et al., 2023; Brown et al., 2020), Computer Vision (Radford et al., 2021; Krizhevsky et al., 2012), Graph Neural Networks (Kipf & Welling, 2017; Rossi et al., 2020). The core idea behind the success of these methods is the ability to construct a mapper $f(x)$ that maps raw data into embeddings high dimension vector. This embedding can be understood as a point in latent space. For example, to encode the semantics of words in the English vocabulary, models learn the representation and embedding of each word in the form of a vector, which can be represented in latent space, Figure 1.

To classify an image as a cat or dog, Convolutional Neural Networks (CNNs) are trained to learn a meaningful presentation of an image in latent space. To predict a property of the chemical compound CH_4, Graph Neural Networks (GNNs) are used to learn the entire structure and properties of a C, H in unified embeddings in latent space, which can be used to infer prediction. Given advanced representation learning, one can use ML models to learn representations of products in e-commerce markets and store these representations in the geometric database, which can later be used as a recommended system (Xu et al., 2020). Motivated by this, the goal of this project is to explore the practical time complexity of two geometric data structures in modeling high-dimensional data.

2 Methodology

We selected two well-known geometric data structures, the KD tree (Bentley, 1975) and the Ball tree (Dolatshah et al., 2015), to explore the practical time complexity of geometric structures in the modeling of high-dimensional data. Both are intended to organize points in higher-dimensional space, with their structure intended to speed up spatial queries such as nearest neighbors. A KD Tree is similar to a binary search tree but in multiple dimensions. At each node, the data is split by the median value of one of the points' dimensions, alternating dimensions with each level. This structure accelerates spatial queries by pruning branches that are too far away. The Ball Tree is similar in the goal of handling multidimensional points, but different in its implementation. It organizes points into nested hyper spheres (ie "balls). It begins by defining one ball around all the points, and then recursively splits the remaining points between two balls until it reaches the defined base case. By checking the ball's boundaries, spatial queries can quickly skip entire sub trees if their ball can not contain a closer point, speeding up traversals. They make sense to compare since they are trying to solve the same problem of efficient spatial searches but are making different trade-offs. The KD Tree works best in lower dimensions and when points are evenly spread out because their axis-aligned splits are easy to compute. However, they struggle in higher dimensions as the "curse of dimensionality" leads to the number of required splits growing exponentially with the number of dimensions. Ball Trees, on the other hand, are more capable of working with higher dimensional data as their sphere shape splits do not rely on-axis alignment. The trade-off to this is that constructing and querying ball trees is more computationally expensive because calculating distances with centroids is more difficult than the KD tree's straight line cuts. Overall, they are directly comparable data structures that are very different in terms of implementation.

[*]Equal contribution

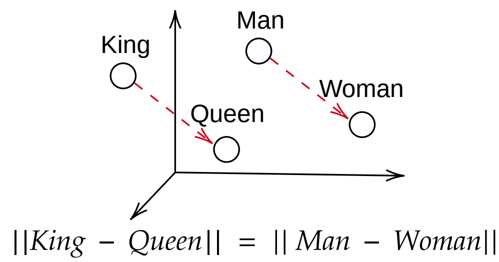

Figure 1: An example of representations of English words. Advancements in ML allow learning representations as high-dimension vectors (embeddings) that reserve semantic information for each word). In this example, the distance and direction of vector $\overrightarrow{KingQueen}$ is relatively the same as vector $\overrightarrow{ManWomman}$

We intend to fully implement KD-tree, Ball tree, and a simple brute force algorithm into a unified framework and test their insert, delete, range search, nearest neighbour, and construction operations. We plan on testing them with both low-dimension data (2D-5D) and higher-dimension data (50D-100D) as well as with sparse and non-sparse data to get a complete picture of their performance. We plan to obtain these data from synthetic data generated by us or from benchmark data used by Huang & Tung.

3 Expected outline

1. Abstract
2. Introduction
3. Related works
4. Notation and Preliminaries
5. Methodology
    (a) KD-tree
    (b) Ball tree
6. Results
7. Discussion

Acknowledgments

This research project uses the ICLR 2021 Conference submission template for formal formatting purposes.

REFERENCES

Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, 1975. doi: 10.1145/361002.361007. URL https://doi.org/10.1145/361002.361007.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Sébastien Bubeck, Varun Chadrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023.

Mohamad Dolatshah, Ali Hadian, and Behrouz Minaei-Bidgoli. Ball*-tree: Efficient spatial indexing for constrained nearest-neighbor search in metric spaces. *CoRR*, abs/1511.00628, 2015. URL http://arxiv.org/abs/1511.00628.

Qiang Huang and Anthony K. H. Tung. Lightweight-yet-efficient: Revitalizing ball-tree for point-to-hyperplane nearest neighbor search. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 436–449. IEEE, 2023.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pp. 1106–1114, 2012. URL https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael M. Bronstein. Temporal graph networks for deep learning on dynamic graphs. *CoRR*, abs/2006.10637, 2020. URL https://arxiv.org/abs/2006.10637.

Da Xu, Chuanwei Ruan, Jason H. D. Cho, Evren Körpeoglu, Sushant Kumar, and Kannan Achan. Knowledge-aware complementary product representation learning. In James Caverlee, Xia (Ben) Hu, Mounia Lalmas, and Wei Wang (eds.), *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*, pp. 681–689. ACM, 2020. doi: 10.1145/3336191.3371854. URL https://doi.org/10.1145/3336191.3371854.