Tutorial on Approximate Nearest Neighbor Search (ANNS) – Techniques and Open Problems

Magdalen Dobson
Carnegie Mellon University
mrdobson@andew.cmu.edu

Laxman Dhulipala
University of Maryland
laxman@umd.edu

Zheqi Shen
UC Riverside
zshen055@ucr.edu

Harsha Vardhan Simhadri
Microsoft Research
harshasi@microsoft.com

1 ANNS AND RECENT DEVELOPMENTS

Approximate nearest neighbor search (ANNS) is a classical algorithmic problem that is increasingly relevant in practice today across a variety of AI application domains. For AI-first applications, ANNS is the key index that connects neural networks for search [27], recommendation [6] and content generation/summarization [10, 18, 19, 25, 26] with relevant items in the knowledge store. This linking process helps better ground the output of the models and generates higher quality results.

While ANNS is a well-studied problem, classical algorithms like LSH [2] and its derivatives fail to meet the requirements of this new generation of applications. Some of the shortcomings include inadequate query performance for large indices of high dimensional vectors (10^7 to 10^{12} vectors with dimensions in the range of $10^2 - 10^4$) and memory-inefficiency and difficulty scaling in the external-memory setting. Perhaps most importantly, existing ANNS approaches are difficult to adapt to new and natural feature requirements motivated by emerging applications. Examples include (a) designing streaming indices, (b) designing ANN indices that can support a combination of hard matches and nearest neighbor search [34], and (c) designing indices that can quickly answer filtered queries.

Addressing these new requirements has been a productive and active new research area, with numerous research papers [4, 15, 20, 22–24, 31, 32] and open search packages [11, 12, 22, 28] aiming to push the research and development frontier. There are also vector-search-as-a-service companies (e.g. Milvus, Pinecone, Qdrant, Vespa, Weaviate, Zilliz) that package this research for commercial use, not to mentioned extensively customized in-house solutions at larger companies where several products stand on ANNS.

2 SCOPE OF THE TUTORIAL

In this 1.5 hour tutorial, we will survey:

- New systems and feature requirements that the next generation of ANNS indices must support to be usable in large-scale deployments.
- Algorithmic techniques developed for ANNS in the last 10 years, and their relative merits and applicability to different use cases. We will focus on graph, and clustering based methods [24, 32] as well as quantization techniques [16, 21].
- Relevant datasets [30], software packages [22, 28], scripts and benchmarking tools [3, 30] for getting started.

An overview of open research problems in generalizing the capabilities of ANN indices and possible directions. For example:

- Complex predicates [34]
- Building updateable vector *databases* with important properties like crash-recovery, serializability and checkpointing
- Designing massive distributed multi-node indices
- Developing better theoretical analysis of empirical algorithms
- Adapting the index as the model parameters that generate the embeddings change, e.g., when ANNS is used inside model training
- Adapting to different query distributions arising, say, a different modality (e.g., text vs image) or from the far tail of the distribution

3 GOALS AND AUDIENCE

Many SPAA attendees have experience in areas relevant to ANNS, such as processing large graphs in parallel and concurrent settings, parallelizing existing ANNS algorithms, and in state-of-the-art data structures for vector databases. The goal of this tutorial is to help such researchers learn about this problem and understand how their expertise relates to open problems in ANNS.

Another goal of the tutorial is to make it easier for new researchers to get started and acquainted with this field. To this end, our tutorial provide a fully-reproducible demonstration of running ANNS on several interesting datasets, showcasing both the power and flexibility of current ANNS systems, as well as their limitations in certain more challenging scenarios, e.g., under dynamic updates or historical queries, in restricted-memory settings, and under filtering, among many others.

Tutorial attendees will run ANNS algorithms using the Problem Based Benchmark Suite (PBBS) [1]. These ANN implementations have been developed using the ParlayLib toolkit [7] and have been highly optimized to yield performance competitive with the original published ANNS. Furthermore, the implementations are provided under the uniform test framework provided in PBBS with easy-to-use interfaces. We will also introduce other existing benchmarking tools for ANNS, such as the ANN-Benchmarks framework [5]. Although this tutorial will focus on relatively small-scale datasets to simplify the problem settings and let the audience focus on the key techniques used in the state-of-the-art ANNS algorithm, the ideas are extended to the billion scale in both the PBBS and the Big ANN Benchmarks framework [29].

4 WHY SPAA?

Approximate nearest neighbor search is a highly parallel problem with interesting algorithmic techniques at its core. The authors of this tutorial have used numerous ideas originating in the SPAA community in ANNS; examples include fundamental parallelism research [7, 8, 17], randomized incremental algorithms [9], and persistent and/or purely functional graph frameworks [13, 14, 33].

Despite its highly parallel nature, there is a lack of researchers whose primary expertise is in parallelism and concurrency working on this problem. One of our main goals in this tutorial, therefore, is to expose the SPAA community to recent developments in ANNS, with a focus on algorithmic techniques used across multiple ANNS systems and open-problems pertaining to these systems. For example, we will pose open questions surrounding finer-grained (potentially input-dependent) analyses of the work, accuracy, and parallelism in widely-deployed ANNS systems.

Furthermore, most of the work on practical ANNS has not provided a theoretical understanding explaining the empirical success of recent ANNS systems. We will also briefly survey the existing theoretical work on ANNS from SPAA and other theory conferences to highlight the current gap between theory and practice.

REFERENCES

[1] Daniel Anderson, Guy E. Blelloch, Laxman Dhulipala, Magdalen Dobson, Yihan Sun, Julian Shun, Jeremy T. Fineman, Phillip B. Gibbons, Aapo Kyrola, Harsha Vardhan Simhadri, and Kanat Tangwongsan. 2023. The Problem Based Benchmark Suite (PBBS). https://github.com/cmuparlay/pbbsbench V2.

[2] Alexandr Andoni, Piotr Indyk, Thijs Laarhoven, Ilya Razenshteyn, and Ludwig Schmidt. 2015. Practical and optimal LSH for angular distance. *Advances in neural information processing systems* 28 (2015).

[3] Martin Aumüller, Erik Bernhardsson, and Alexander Faithfull. 2020. ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. *Information Systems* 87 (2020). http://www.sciencedirect.com/science/article/pii/S0306437918303685

[4] Artem Babenko and Victor Lempitsky. 2012. The inverted multi-index. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 3069–3076. https://doi.org/10.1109/CVPR.2012.6248038

[5] Erik Bernhardsson, Martin Aumüller, and Alexander Faithfull. 2019. ANN Benchmarks. http://ann-benchmarks.com/

[6] K. Bhatia, K. Dahiya, H. Jain, P. Kar, A. Mittal, Y. Prabhu, and M. Varma. 2016. The extreme classification repository: Multi-label datasets and code. http://manikvarma.org/downloads/XC/XMLRepository.html

[7] Guy E. Blelloch, Daniel Anderson, and Laxman Dhulipala. 2020. ParlayLib - A Toolkit for Parallel Algorithms on Shared-Memory Multicore Machines. In *SPAA '20: 32nd ACM Symposium on Parallelism in Algorithms and Architectures, Virtual Event, USA, July 15-17, 2020*. ACM, 507–509.

[8] Guy E. Blelloch, Jeremy T. Fineman, Yan Gu, and Yihan Sun. 2020. Optimal Parallel Algorithms in the Binary-Forking Model. In *SPAA '20: 32nd ACM Symposium on Parallelism in Algorithms and Architectures, Virtual Event, USA, July 15-17, 2020*, Christian Scheideler and Michael Spear (Eds.). ACM, 89–102. https://doi.org/10.1145/3350755.3400227

[9] Guy E. Blelloch, Yan Gu, Julian Shun, and Yihan Sun. 2016. Parallelism in Randomized Incremental Algorithms. In *Proceedings of the 28th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA 2016, Asilomar State Beach/Pacific Grove, CA, USA, July 11-13, 2016*. ACM, 467–478.

[10] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. arXiv:2112.04426 [cs.CL]

[11] Leonid Boytsov, Bilegsaikhan Naidan, Yury Malkov, et al. 2023. Non-Metric Space Library (NMSLIB). https://github.com/nmslib/nmslib

[12] Qi Chen, Bing Zhao, Haidong Wang, Mingqin Li, Chuanjie Liu, Zengzhong Li, Mao Yang, and Jingdong Wang. 2021. SPANN: Highly-efficient Billion-scale Approximate Nearest Neighbor Search. *CoRR* abs/2111.08566 (2021). arXiv:2111.08566 https://arxiv.org/abs/2111.08566

[13] Laxman Dhulipala, Guy E. Blelloch, Yan Gu, and Yihan Sun. 2022. PaC-trees: supporting parallel and compressed purely-functional collections. In *PLDI '22: 43rd ACM SIGPLAN International Conference on Programming Language Design and Implementation, San Diego, CA, USA, June 13 - 17, 2022*. ACM, 108–121.

[14] Laxman Dhulipala, Guy E. Blelloch, and Julian Shun. 2019. Low-latency graph streaming using compressed purely-functional trees. In *Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation, PLDI 2019, Phoenix, AZ, USA, June 22-26, 2019*. ACM, 918–934.

[15] Cong Fu, Chao Xiang, Changxu Wang, and Deng Cai. 2019. Fast Approximate Nearest Neighbor Search With The Navigating Spreading-out Graphs. *PVLDB* 12, 5 (2019), 461 – 474. https://doi.org/10.14778/3303753.3303754

[16] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. 2014. Optimized Product Quantization. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 4 (2014), 744–755. https://doi.org/10.1109/TPAMI.2013.240

[17] Yan Gu, Julian Shun, Yihan Sun, and Guy E. Blelloch. 2015. A Top-Down Parallel Semisort. In *Proceedings of the 27th ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2015, Portland, OR, USA, June 13-15, 2015*. ACM, 24–34.

[18] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. REALM: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909* (2020).

[19] Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2022. Few-shot Learning with Retrieval Augmented Language Models. (2022). http://arxiv.org/abs/2208.03299

[20] Shikhar Jaiswal, Ravishankar Krishnaswamy, Ankit Garg, Harsha Vardhan Simhadri, and Sheshansh Agrawal. 2022. OOD-DiskANN: Efficient and Scalable Graph ANNS for Out-of-Distribution Queries. https://doi.org/10.48550/ARXIV.2211.12850

[21] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. 2011. Product Quantization for Nearest Neighbor Search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 1 (Jan. 2011), 117–128. https://doi.org/10.1109/TPAMI.2010.57

[22] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. *arXiv preprint arXiv:1702.08734* (2017).

[23] Yu Malkov, Alexander Ponomarenko, Andrey Logvinov, and Vladimir Krylov. 2013. Approximate nearest neighbor algorithm based on navigable small world graphs. *Information Systems* 45 (01 2013), 61–68. https://doi.org/10.1016/j.is.2013.10.006

[24] Yury A. Malkov and D. A. Yashunin. 2016. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. *CoRR* abs/1603.09320 (2016). arXiv:1603.09320 http://arxiv.org/abs/1603.09320

[25] Yusuf Mehdi. 2023. Reinventing search with a new AI-powered Microsoft Bing and Edge, your copilot for the web. https://blogs.microsoft.com/blog/2023/02/07/reinventing-search-with-a-new-ai-powered-microsoft-bing-and-edge-your-copilot-for-the-web/

[26] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332* (2021).

[27] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. *choice* 2640 (2016), 660.

[28] Harsha Vardhan Simhadri, Ravishankar Krishnaswamy, Gopal Srinivasa, Suhas Jayaram Subramanya, Andrija Antonijevic, Dax Pryce, David Kaczynski, Shane Williams, Siddarth Gollapudi, Varun Sivashankar, Neel Karia, Aditi Singh, Shikhar Jaiswal, Neelam Mahapatro, Philip Adams, and Bryan Tower. 2023. DiskANN: Scalable, Efficient and Feature-rich Approximate Nearest Neighbor Search. https://github.com/Microsoft/DiskANN Version 0.5.

[29] Harsha Vardhan Simhadri, George Williams, Martin Aumüller, Matthijs Douze, Artem Babenko, Dmitry Baranchuk, Qi Chen, Lucas Hosseini, Ravishankar Krishnaswamy, Gopal Srinivasa, Suhas Jayaram Subramanya, and Jingdong Wang. 2021. Billion-Scale Approximate Nearest Neighbor Search Challenge: NeurIPS'21 competition track. https://big-ann-benchmarks.com/

[30] Harsha Vardhan Simhadri, George Williams, Martin Aumüller, Matthijs Douze, Artem Babenko, Dmitry Baranchuk, Qi Chen, Lucas Hosseini, Ravishankar Krishnaswamy, Gopal Srinivasa, Suhas Jayaram Subramanya, and Jingdong Wang. 2022. Results of the NeurIPS'21 Challenge on Billion-Scale Approximate Nearest Neighbor Search. https://doi.org/10.48550/ARXIV.2205.03763

[31] Aditi Singh, Suhas Jayaram Subramanya, Ravishankar Krishnaswamy, and Harsha Vardhan Simhadri. 2021. FreshDiskANN: A Fast and Accurate Graph-Based ANN Index for Streaming Similarity Search. *CoRR* abs/2105.09613 (2021). arXiv:2105.09613 https://arxiv.org/abs/2105.09613

[32] Suhas Jayaram Subramanya, Fnu Devvrit, Rohan Kadekodi, Ravishankar Krishnawamy, and Harsha Vardhan Simhadri. 2019. DiskANN: Fast Accurate Billion-point Nearest Neighbor Search on a Single Node. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). 13748–13758. http://papers.nips.cc/paper/9527-rand-nsg-fast-accurate-billion-point-nearest-neighbor-search-on-a-single-node

[33] Yihan Sun, Daniel Ferizovic, and Guy E. Blelloch. 2018. PAM: parallel augmented maps. In *Proceedings of the 23rd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, PPoPP 2018, Vienna, Austria, February 24-28, 2018*. ACM, 290–304.

[34] Chuangxian Wei, Bin Wu, Sheng Wang, Renjie Lou, Chaoqun Zhan, Feifei Li, and Yuanzhe Cai. 2020. AnalyticDB-V: A Hybrid Analytical Engine Towards

Query Fusion for Structured and Unstructured Data. *Proc. VLDB Endow.* 13, 12 (2020), 3152–3165. https://doi.org/10.14778/3415478.3415541