High-Dimensional Similarity Query Processing for Data Science [Tutorial Proposal]

Jianbin Qin
Shenzhen University & Shenzhen Institute of Computing Sciences
Shenzhen, Guangdong, China
qinjianbin@szu.edu.cn
+86-755-26532350

Wei Wang
University of New South Wales & Dongguan University of Technology
Kensington, NSW, Australia
weiw@cse.unsw.edu.au
+61-2-9385-7162

Chuan Xiao
Osaka University & Nagoya University
Suita, Osaka, Japan
chuanx@ist.osaka-u.ac.jp
+81-6-6105-6502

Ying Zhang
University of Technology Sydney
Ultimo, NSW, Australia
Ying.Zhang@uts.edu.au
+61-2-9514-1103

Yaoshu Wang
Shenzhen University & Shenzhen Institute of Computing Sciences
Shenzhen, Guangdong, China
yaoshuw@sics.ac.cn
+86-755-26532350

ABSTRACT

Similarity query (a.k.a. nearest neighbor query) processing has been an active research topic for several decades. It is an essential procedure in a wide range of applications (e.g., classification & regression, deduplication, image retrieval, and recommender systems). Recently, representation learning and auto-encoding methods as well as pre-trained models have gained popularity. They basically deal with dense high-dimensional data, and this trend brings new opportunities and challenges to similarity query processing. Meanwhile, new techniques have emerged to tackle this long-standing problem theoretically and empirically.

This tutorial aims to provide a comprehensive review of high-dimensional similarity query processing for data science. It introduces solutions from a variety of research communities, including data mining (DM), database (DB), machine learning (ML), computer vision (CV), and theoretical computer science (TCS), thereby highlighting the interplay between modern DM, DB, ML, CV, and TCS technologies. We first discuss the importance of high-dimensional similarity query processing in data science applications, and then review query processing algorithms such as cover tree, locality sensitive hashing, product quantization, proximity graphs, as well as recent advancements such as learned indexes. We analyze their strengths and weaknesses and discuss the selection of algorithms in various application scenarios. Moreover, we consider the selectivity estimation of high-dimensional similarity queries, and show how researchers are bringing in state-of-the-art ML techniques to address this problem. We expect that this tutorial will provide an impetus towards new technologies for data science.

1 TARGET AUDIENCE

This tutorial targets researchers, developers, and practitioners interested in data science. We assume that the target audience is generally familiar with basic data mining, database, and machine learning terms, but there is no requirement for prior knowledge of specific algorithms.

2 TUTORS

Please refer to the author list at the top of this page for **names, affiliations, addresses, email addresses, and phone numbers**.

Presenter(s):

- **Jianbin Qin** [1] is a Professor with Shenzhen University and a Research Scientist with Shenzhen Institute of Computing Sciences. He received the Ph.D. degree from the University of New South Wales in 2013. His research interests include similarity query processing, data integration, textual databases, and information retrieval. His work with Wei Wang and Chuan Xiao – presenters of this tutorial – on similarity query processing has been selected among the best papers of SIGMOD 2011 and ICDE 2018. He has given tutorials at VLDB 2020 and WISE 2017.
- **Wei Wang** [2] is a Professor with the University of New South Wales and Dongguan University of Technology. He received the Ph.D. degree from the Hong Kong University of Science and Technology in 2004. His

[1] https://jqin.gitee.io/
[2] http://www.cse.unsw.edu.au/~weiw/

research interests include high-dimensional data management, similarity query processing, data integration, knowledge graphs, natural language processing, and adversarial machine learning. He has given tutorials at ICDE 2011, SIGMOD 2009, and VLDB 2020.

- **Chuan Xiao** [3] is an Associate Professor with Osaka University and Nagoya University. He received the Ph.D. degree from the University of New South Wales in 2010. His research interests include similarity query processing, data integration, spatio-temporal databases, and information retrieval. He has given tutorials at VLDB 2020 and WISE 2017.
- **Ying Zhang** [4] is a Professor and ARC Future Fellow with the University of Technology Sydney, and the Head of the Database Group at the Centre for Artificial Intelligence. He received the Ph.D. degree from the University of New South Wales in 2008. His research interests include high-dimensional data management, scalable data analytics, data streams, and graph databases. He has given tutorials at ICDE 2019 and VLDB 2020.

Contributor(s):

- **Yaoshu Wang** [5] is a Research Scientist with Shenzhen University and Shenzhen Institute of Computing Sciences. He received the Ph.D. degree from the University of New South Wales in 2018. His research interests include similarity query processing, data integration, and textual databases.

The tutors have rich experience in the research on similarity query processing and have made significant contributions to this topic at top venues (e.g., [15, 17, 21, 24, 25]). The presenters have given a tutorial on this topic at VLDB 2020.

3 CORRESPONDING TUTOR

Chuan Xiao (chuanx@ist.osaka-u.ac.jp).

4 TUTORIAL OUTLINE

This tutorial consists of five parts. The first part motivates the need for high-dimensional similarity query processing and introduces basic concepts. The second and third parts delve into exact and approximate query processing algorithms, respectively. The fourth part covers selectivity estimation algorithms. The fifth part concludes the tutorial by discussing future directions and open problems.

[3] https://sites.google.com/site/chuanxiao1983/
[4] https://www.uts.edu.au/staff/ying.zhang
[5] https://en.sics.ac.cn/people/yaoshu-wang

4.1 Background and Preliminaries

In the introductory part of the tutorial, we first introduce the applications of high-dimensional similarity query processing in data science (e.g., in anomaly detection [11] and recommender systems [6]) and explains its increasing importance. Then we describe basic concepts: (1) data models and the way of which we convert raw data (text, images, video, etc.) to high-dimensional data; (2) similarity/distance functions, mainly Hamming distance for binary vectors and Euclidean distance and cosine similarity (angular distance) for real-valued vectors; (3) query types, i.e., search and join queries, or thresholded and top-k (k-NN) queries, depending on the dimension of categorization; (4) a summary of the solutions that will be elaborated in the rest of the tutorial.

4.2 Exact Query Processing

Exact query processing methods aim to find all the results that satisfy the similarity constraint. Researchers are interested in this type of solutions as it does not pose any uncertainty to the pipelines that apply similarity query processing as a component. It also simplifies empirical comparison as only speed and space consumptions are key evaluation criteria. Existing exact methods can be classified into the following three categories:

Tree-based Methods. These methods partition the set of data points in a hierarchical manner. To process queries, triangle inequality is often used to determine the nodes to be traversed (e.g., cover tree [13]).

Space Partitioning Methods. These methods partition the original space and bound the overall distance using the distance within each part. Some methods require a sequential scan of the data points, e.g., the vector approximation file (VA-file) [26]. For fast retrieval, indexing methods were proposed to deal with Hamming distance using the pigeonhole principle [20, 21].

Dimensionality Reduction Methods. These methods transform data points to another space to reduce dimensionality. They are basically early attempts that deal with the disk-resident case and aim at reducing disk I/O [2, 5]. Most of them transform the original space to a 1-dimensional space and utilize B^+-trees for indexing.

4.3 Approximate Query Processing

It is commonly believed that it is hard to compute the exact results of queries with a sub-linear cost due to the curse of dimensionality. Instead, computing approximate results is sufficiently useful for many practical problems, and these solutions empirically achieve significantly higher efficiency and scalability than exact ones.

Locality Sensitive Hashing. Locality sensitive hashing (LSH) is a data-independent hashing approach with probabilistic

guarantees on the worst-case performance [12]. It relies on a family of hash functions that map similar data points to the same hash codes with higher probability than dissimilar points. Plenty of solutions have been proposed. Recent development focuses on supporting various similarity measures [30] and space-efficient indexing [24, 29].

Learning to Hash. Learning to Hash (L2H) methods maps original data to another (often Hamming) space by exploiting the data distribution. The underlying principle is to preserve the similarity information within an appropriate neighborhood. Additional heuristics and optimizations are often added to further reduce the information loss caused by the mapping or increase generalization to unseen data. Recent advancements in L2H methods feature deep learning in both supervised and unsupervised manner [4, 10, 16, 23]. Another line of methods is based on product quantization [14], with the unique ability to handle billions of data points. Due to its excellent scalability, there are many extensions in optimizing its coding scheme, indexing, and searching.

Partition-based Methods. Methods in this category can be deemed as dividing the high-dimensional space into multiple disjoint regions. Partition is often carried out in a recursive way, so the index is represented by a tree or a forest. Based on the way of partitioning, there are mainly three classes of methods: Pivoting methods divide the data points based on the distance from the point to some (usually randomly chosen) pivots (e.g., VP-Tree [28]). Hyperplane partitioning methods recursively divide the space by a hyperplane with a random direction (e.g., Annoy [3] and random projection tree [7]) or an axis-aligned separating hyperplane (e.g., kd-tree [22]). Compact partitioning methods either divide the data points into clusters or create possibly approximate Voronoi partitions to exploit locality.

Neighborhood-based Methods. Graph-based methods construct a proximity graph where nodes represent data points and edges connect nearby points. The main idea is to perform a search for similar data points atop the proximity graph. These methods achieve top accuracy and speed trade-off in many empirical evaluations [15]. The first class of these methods tries to build a k-NN graph [8] which records the top-k nearest neighbors of each point. Then nearest neighbor search is conducted by the hill-climbing strategy. The second class employs the navigable small world graph, an undirected graph that contains an approximation of the Delaunay graph and has long-range links with the small world navigation property. Hierarchical navigable small world [18] is one of the most efficient algorithms thus far and can support incremental update. Recently, learned indexes were proposed to provide a more efficient search path in the graph [1]. The third class is based on the relative neighborhood graph [9], which considers connectivity, degree, shortest path length, and index size for graph construction. It was shown to achieve more robust empirical performance.

4.4 Selectivity Estimation

Selectivity estimation outputs the approximate number of data objects that satisfy a selection criterion. Due to its use in density estimation, outlier detection, image retrieval, and query optimization, selectivity estimation for high-dimensional data has received considerable attention recently. For example, hands-off entity matching systems extract paths from random forests and take each path (a conjunction of similarity predicates over multiple attributes) as a blocking rule, and thus selectivity estimation is useful for choosing the execution order of query plans that involve multiple similarity predicates. Although selectivity estimation for queries on relational data has been extensively studied in the DB community, few of these methods are applicable to high-dimensional data due to the curse of dimensionality. Representative solutions to this problem are sampling [27] and kernel density estimation [19]. A recent trend is to formalize it as a regression task and utilize ML methods [25].

4.5 Future Opportunities

We highlight a number of promising directions for future research: (1) It is interesting to explore ML models as approximate solutions to search queries (e.g., learning to index or learning to sample). (2) Answering composite queries (e.g., conjunctive queries) over multiple attributes will receive more attention, since many data science tasks deal with multi-attribute data and the advancement of ML methods will enable us to convert multi-attribute data to distributed representations for semantic comparison. (3) Another direction is to develop efficient algorithms for query processing in end-to-end systems, where ML and natural language processing techniques can improve the quality.

5 PREVIOUS EDITIONS

The previous edition of this tutorial, entitled "Similarity Query Processing for High-Dimensional Data", appeared at VLDB 2020 [6] (virtual conference, Aug 31 – Sep 4, 2020) and was live-presented via Zoom.

The new edition is a tutorial focused on **data science related applications** (e.g., classification, regression, anomaly detection, and recommender systems). In addition, the new edition features the following new materials:

(1) a thorough discussion on the use of similarity query processing in various application scenarios (e.g., the role of similarity queries in the entire workflow, the selection of algorithms, and the solution to billion scale),

[6] Presentation slides are available at http://www.cse.unsw.edu.au/~weiw/resources/vldb20-tutorial-simqp-high-dim/

(2) more data models and a broader range of related works, and
(3) more recent technical advancements (e.g., learned indexes) and future trends.

In addition, three topically related tutorials were presented at

(1) WISE 2017 (Moscow, October 7 – 11, 2017), "Set Similarity Query Processing",
(2) CIKM 2019 (Beijing, November 3 – 7, 2019), "Synergy of Database Techniques and Machine Learning Models for String Similarity Search and Join", and
(3) CVPR 2020 (virtual conference, June 14 – 19, 2020), "Image Retrieval in the Wild".

Albeit related to similarity query processing, the WISE 2017 and CIKM 2019 tutorials target set and string similarity, respectively, whose query processing methods are substantially different from those will be presented at our tutorial. The CVPR 2020 tutorial has a focus on image retrieval and briefly introduces some high-dimensional query processing methods as part of the program, while our tutorial will cover more application scenarios, data models, query processing methods, and a dedicated part to selectivity estimation.

6 LIST OF REFERENCES

Please see the references at the end of this proposal.

7 INTERACTION STYLE

This tutorial follows a lecture style. In each part of the tutorial, the tutors present the lecturing materials (in presentation slides) and interact with the audience through a Q&A session. Examples in real-world data science applications will be used to explain the role of high-dimensional similarity queries in the entire workflow. Solutions will be compared in a thorough manner. To help understand the selection of algorithms in different application scenarios, we will provide a cheat sheet so that the audience can quickly get the take-away of the discussion.

8 EQUIPMENT

- Equipment the presenters will bring: a laptop.
- Equipment the presenters will need: a pointer and power sockets.
- Equipment attendees should bring: none.

9 TUTORIAL WEBSITE

Slides, bibliography, and other related materials will be available at the following URL:

http://www.cse.unsw.edu.au/~weiw/resources/

REFERENCES

[1] D. Baranchuk and A. Babenko. Towards similarity graphs constructed by deep reinforcement learning. *CoRR*, abs/1911.12122, 2019.
[2] S. Berchtold, C. Böhm, and H. Kriegel. The pyramid-technique: Towards breaking the curse of dimensionality. In *SIGMOD*, pages 142–153, 1998.
[3] E. Bernhardsson. Annoy at github https://github.com/spotify/annoy, 2015.
[4] D. Cai, X. Gu, and C. Wang. A revisit on deep hashings for large-scale content based image retrieval. *CoRR*, abs/1711.06016, 2017.
[5] K. Chakrabarti and S. Mehrotra. Local dimensionality reduction: A new approach to indexing high dimensional spaces. In *VLDB*, pages 89–100, 2000.
[6] P. Covington, J. Adams, and E. Sargin. Deep neural networks for youtube recommendations. In *RecSys*, pages 191–198, 2016.
[7] S. Dasgupta and Y. Freund. Random projection trees and low dimensional manifolds. In *STOC*, pages 537–546, 2008.
[8] W. Dong, M. Charikar, and K. Li. Efficient k-nearest neighbor graph construction for generic similarity measures. In *WWW*, pages 577–586, 2011.
[9] C. Fu, C. Xiang, C. Wang, and D. Cai. Fast approximate nearest neighbor search with the navigating spreading-out graph. *PVLDB*, 12(5):461–474, 2019.
[10] N. Gao, M. Wilson, T. Vandal, W. Vinci, R. R. Nemani, and E. G. Rieffel. High-dimensional similarity search with quantum-assisted variational autoencoder. In *KDD*, pages 956–964, 2020.
[11] X. Gu, L. Akoglu, and A. Rinaldo. Statistical analysis of nearest neighbor methods for anomaly detection. In *NeurIPS*, pages 10921–10931, 2019.
[12] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *STOC*, pages 604–613, 1998.
[13] M. Izbicki and C. R. Shelton. Faster cover trees. In *ICML*, pages 1162–1170, 2015.
[14] H. Jégou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):117–128, 2011.
[15] W. Li, Y. Zhang, Y. Sun, W. Wang, M. Li, W. Zhang, and X. Lin. Approximate nearest neighbor search on high dimensional data - experiments, analyses, and improvement. *IEEE Trans. Knowl. Data Eng.*, 32(8):1475–1488, 2020.
[16] J. Lu, V. E. Liong, and J. Zhou. Deep hashing for scalable image search. *IEEE Trans. Image Processing*, 26(5):2352–2367, 2017.
[17] K. Lu, H. Wang, W. Wang, and M. Kudo. VHP: approximate nearest neighbor search via virtual hypersphere partitioning. *PVLDB*, 13(9):1443–1455, 2020.
[18] Y. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(4):824–836, 2020.
[19] M. Mattig, T. Fober, C. Beilschmidt, and B. Seeger. Kernel-based cardinality estimation on metric data. In *EDBT*, pages 349–360, 2018.
[20] M. Norouzi, A. Punjani, and D. J. Fleet. Fast exact search in hamming space with multi-index hashing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(6):1107–1119, 2014.
[21] J. Qin, C. Xiao, Y. Wang, W. Wang, X. Lin, Y. Ishikawa, and G. Wang. Generalizing the pigeonhole principle for similarity search in hamming space. *IEEE Trans. Knowl. Data Eng.*, 33(2):489–505, 2021.
[22] P. Ram and K. Sinha. Revisiting kd-tree for nearest neighbor search. In *KDD*, pages 1378–1388, 2019.
[23] F. Shen, Y. Xu, L. Liu, Y. Yang, Z. Huang, and H. T. Shen. Unsupervised deep hashing with similarity-adaptive and discrete optimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):3034–3044, 2018.

[24] Y. Sun, W. Wang, J. Qin, Y. Zhang, and X. Lin. SRS: solving c-approximate nearest neighbor queries in high dimensional euclidean space with a tiny index. *PVLDB*, 8(1):1–12, 2014.

[25] Y. Wang, C. Xiao, J. Qin, X. Cao, Y. Sun, W. Wang, and M. Onizuka. Monotonic cardinality estimation of similarity selection: A deep learning approach. In *SIGMOD*, pages 1197–1212, 2020.

[26] R. Weber, H. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB*, pages 194–205, 1998.

[27] X. Wu, M. Charikar, and V. Natchu. Local density estimation in high dimensions. In *ICML*, pages 5293–5301, 2018.

[28] P. N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *SODA*, pages 311–321, 1993.

[29] B. Zheng, X. Zhao, L. Weng, N. Q. V. Hung, H. Liu, and C. S. Jensen. PM-LSH: A fast and accurate LSH framework for high-dimensional approximate NN search. *PVLDB*, 13(5):643–655, 2020.

[30] E. Zhu, F. Nargesian, K. Q. Pu, and R. J. Miller. LSH ensemble: Internet-scale domain search. *PVLDB*, 9(12):1185–1196, 2016.