New Trends in High-D Vector Similarity Search: AI-driven, Progressive, and Distributed

Karima Echihabi
Mohammed VI Polytechnic University
karima.echihabi@um6p.ma

Kostas Zoumpatianos
Harvard University
kostas@seas.harvard.edu

Themis Palpanas
Université de Paris & IUF
themis@mi.parisdescartes.fr

ABSTRACT

Similarity search is a core operation of many critical applications, involving massive collections of high-dimensional (high-d) objects. Objects can be data series, text, multimedia, graphs, database tables or deep network embeddings. In this tutorial, we revisit the similarity search problem in light of the recent advances in the field and the new big data landscape. We discuss key data science applications that require efficient high-d similarity search, we survey recent approaches and share surprising insights about their strengths and weaknesses, and we discuss open research problems, including the directions of AI-driven, progressive, and distributed high-d similarity search.

PVLDB Reference Format:
Karima Echihabi, Kostas Zoumpatianos, and Themis Palpanas. New Trends in High-D Vector Similarity Search:
AI-driven, Progressive, and Distributed. PVLDB, 14(12): 3198 - 3201, 2021.
doi:10.14778/3476311.3476407

1 INTRODUCTION

Similarity search aims at finding objects in a collection that are close to a given query according to some definition of sameness. It is a fundamental operation that lies at the core of many critical data science applications. In data integration, it has been used to automate entity resolution [20] and support data discovery [76]. It has powered electricity demand analytics [43], recommender systems of online billion-dollar enterprises [66] and enabled clustering [12], classification [57] and outlier detection [10, 13, 46] in domains as varied as bioinformatics, computer vision, security, finance and medicine. Similarity search has also been exploited in software engineering [3] to automate API mappings and predict program dependencies, and in cybersecurity to profile network usage and detect malware [19].

This problem has been studied heavily in the past 25 years and will continue to attract attention as massive collections of high-dimensional (high-d) objects are becoming omnipresent [23, 51, 53]. Objects can be data series, text, images, audio and video recordings, graphs, database tables or deep network embeddings. Similarity search over high-d objects is often reduced to a k-Nearest Neighbor (k-NN) problem such that the objects are represented using high-d vectors and the (dis)-similarity between them is measured using a distance. Some studies [1, 9] have argued that NN search is not meaningful for a number of high-d datasets due to the concentration of distances (a.k.a. the curse of dimensionality). However, these conclusions were based on over-restrictive assumptions such as data being identical and independently distributed (i.i.d.) in each dimension, dimensionality being the only factor determining meaningfulness and an asymptotic analysis of dimensionality growing to infinity. In fact, other studies have shown that high-d NN search is meaningful for non-i.i.d data, data with low intrinsic dimensionality and for a variety of real world datasets [35]. The importance and relevance of NN search in high-d is further evidenced by a large and growing body of research [21].

High-d similarity search is hard, because objects often contain 100s-1000s of dimensions. For large datasets, the cost to compare a query to all objects in the collection becomes prohibitive both in terms of CPU and I/O. Similarity search algorithms can either return exact or approximate answers. Exact methods are expensive while approximate methods sacrifice accuracy to achieve better efficiency. We call methods that provide no guarantees on the results *ng*-approximate, and those supporting guarantees on the approximation error, δ-ϵ-approximate methods, where ϵ is the approximation error and δ, the probability that ϵ will not be exceeded. When $\delta = 1$, a δ-ϵ-approximate method becomes ϵ-approximate, and when $\epsilon = 0$, an ϵ-approximate method becomes exact.

This tutorial covers data science applications requiring efficient high-d similarity search, provides an overview of the state-of-the-art exact and approximate high-d similarity search approaches, and discusses the open research problems in this domain, including the directions of progressive and distributed solutions, as well as solutions that integrate Artificial Intelligence (AI).

2 TRENDS AND CHALLENGES

Similarity search has been studied in the past 25 years by different communities often using diverse and conflicting terminology. We present a unified terminology and a taxonomy (Fig 1; non-exhaustive) for similarity search techniques [26, 27], in order to facilitate further work in this area.

[Exact Search] Exact techniques guarantee correct results at the expense of efficiency and footprint. The research community has developed exact approaches for generic high-d vectors [8, 16, 29, 70] (for exhaustive surveys, see [26, 60]).

Exact similarity search methods can be classified into sequential and indexing methods. Sequential methods answer a similarity search query in one phase, reading each candidate sequentially from the raw data file and comparing it to the query. Indexes use a filter and refine approach to answer a similarity query. A pre-built index is used to filter candidates, which are then compared to the query in the raw high-d space for refinement [29, 34, 52, 69, 70, 77]. We note that all indexing methods depend on lower-bounding,

Proceedings of the VLDB Endowment, Vol. 14, No. 12 ISSN 2150-8097.
doi:10.14778/3476311.3476407

which allows indexes to prune the search space with the guarantee of no false dismissals [28].

[Approximate Search] Since exact similarity search is expensive, approximate techniques have been proposed to improve search efficiency at the expense of accuracy. The key research problem in approximate search is making the right trade-offs between accuracy, efficiency and footprint.

Approximate Search With Guarantees. δ-ϵ-approximate search dates from 1998 [37] and gave rise to a rich family of LSH algorithms [67], which solve the problem in sub-linear time, for $\delta < 1$. The main idea is that two neighbors in a high-d space will remain in close proximity when projected to a lower dimensional space. There exist many variants of LSH, either proposing different hash functions to support particular similarity measures [11, 17, 31], or improving the theoretical bounds on query accuracy (i.e., δ or ϵ), query efficiency or the index size [36, 63].

A δ-ϵ-approximate search algorithm was also proposed for the MTree [15], and the same ideas were used to extend existing exact data series techniques to enable them to support δ-ϵ-approximate search [27]. These extensions outperformed the MTree and the state-of-the-art LSH techniques [36, 63] across the board in efficiency, accuracy and footprint, in-memory and on-disk, using real and synthetic datasets.

Approximate Search Without Guarantees. As LSH-based techniques require high footprint and are considered slow for many applications, *ng*-approximate methods that sacrifice guarantees all together were proposed to provide answers faster with good empirical accuracy. The most popular methods in this class are neighborhood graphs [18, 30, 49] and inverted indexes [6, 32, 39]. HNSW [6, 49], a proximity method based on navigable small world graphs, is considered the best contender for in-memory *ng*-approximate search [5, 27, 45], while data series similarity search methods have superior performance on-disk [27].

The practicality of *ng*-approximate similarity search will be further enhanced by improving the footprint and indexing efficiency of existing neighborhood-based methods, and designing new techniques that scale to disk-based data [38].

[Progressive Search] Although the recent state-of-the-art exact techniques push the efficiency frontier, we observe that their query answering times are still not satisfactory for interactive analytics. A promising research direction is to equip exact algorithms with progressive query answering so that they return progressive estimates of the final answer with probability guarantees supporting interactive exploration [65, 74].

We will demonstrate the importance of providing progressive similarity search results on large high-d vector collections. For exact search in particular, there is a gap between the time the 1st Nearest Neighbor (1-NN) is found and the time when the search algorithm terminates, which means that users often wait without any improvement in their answers. In some cases, high-quality approximate answers are found very early, e.g., in less than one second, so they can support highly interactive visual analysis tasks [33].

Similar observations are also true for approximate search. Li et al. [44] propose a machine learning method, developed on top of an inverted-file (IVF [39] and IMI [6, 49]) and a k-NN graph (HNSW [6, 49]) similarity search techniques, that solves the problem of early termination of approximate NN queries, while achieving a target recall.

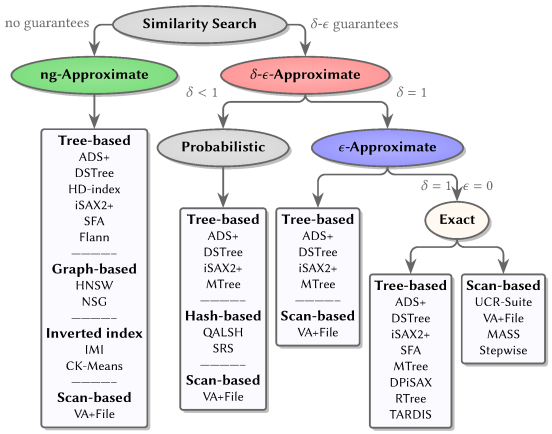

Figure 1: Taxonomy of similarity search methods.

[Revisiting Guarantees] We observe that popular *ng*-approximate techniques may return incomplete result sets, e.g., retrieving only a subset of the neighbors for a k-NN query, yet establishing guarantees on search results is important for several applications [53]. Techniques that offer guarantees, focus on two dimensions that relate to data quality: query accuracy and answering time. Key future directions in this area are extending new types of guarantees and developing new cost models. These will also be relevant for the efforts on progressive similarity search.

In the approximate search literature, query accuracy has been evaluated using recall, and approximation error. LSH techniques are considered the state-of-the-art in approximate search with theoretically proven sublinear time performance and probabilistic guarantees on accuracy (approximation error) [47]. Recent results though, indicate that using the approximate search functionality of data series techniques provides tighter bounds than LSH and a much better performance in practice, with experimental accuracy levels well above the theoretical accuracy guarantees [27]. Note that LSH techniques can only provide probabilistic answers ($\delta < 1$), whereas the extended data series methods can also answer exact and ϵ-approximate queries ($\delta = 1$). A promising research direction is to improve the existing guarantees, or establish new ones: (1) adding guarantees on query time performance; (2) developing probabilistic or deterministic guarantees on the recall or MAP value of a result set, instead of the commonly used distance approximation error. It has been demonstrated that Recall and MAP are better indicators of accuracy, because even small approximation errors may still result in low recall/MAP values [4, 27].

[AI-driven Similarity Search] Data Representation. Similarity search methods rely on dimensionality reduction to achieve efficiency. Some works proposed learned hashed functions [14, 48] while others introduced learned quantization techniques [50, 71]. It would be worthwhile to further explore how machine learning algorithms can improve dimensionality reduction techniques [68], and tailor them to datasets from various domains (data series, images, deep network embeddings). It would then be critical to establish

lower-bounds for these learned summarizations so that they can be exploited by indexes.

Search and Indexing. Machine learning techniques have been leveraged to build indexing structures for similarity search, including kNN graphs [62] and multidimensional indexes [2]. Other works have focused on improving search performance and accuracy [58].

Approximate similarity search techniques based on data series indexes [27] are very practical because they build the index once and tune the desired accuracy/efficiency tradeoff at query time. A promising research direction is to exploit AI to learn more effective stopping conditions, which can further improve the efficiency of these techniques. Another interesting direction is to build upon recent results for data distribution estimation [64], in order to facilitate query answering over high-d datasets.

Performance Tuning. Tuning most approximate similarity search techniques based on LSH, k-NN graphs and inverted indexes is cumbersome and time-consuming [27]. For instance, QALSH [36] needs to build a different index for each desired query accuracy. This is a serious drawback that concerns also IMI and HSNW, which are regarded among the best *ng*-approximate methods. The fact that the speed-accuracy tradeoff depends not only on query answering, but also on index building, means that an index may need to be built many times, using different parameters, before finding the right speed-accuracy tradeoff. Besides, the optimal settings may differ across datasets, or different dataset sizes for the same dataset. Developing auto-tuning methods for these techniques is both an interesting problem and a necessity.

[Systems Considerations] Modern Hardware and Distribution. A number of similarity search techniques have been proposed for modern hardware, focusing on SIMD, multi-socket architectures, the GPU and SSD storage [42, 54–56], as well as on distributed architectures [7, 72, 73, 75], which we will present and discuss. Interesting directions include the development of methods for advanced technologies, such as FPGA and NVM.

End-to-end Solutions While most studies have focused on the high-d similarity search problem from an algorithmic point of view, more effort should go into building end-to-end systems that provide native support for high-d vectors, including similarity search, which is the basis for building complex analytics. There is a significant effort under way in the context of data series [40], though, more advanced and general systems are needed.

[Benchmarks] Despite the importance of benchmarking for evaluating the performance of existing solutions and identifying opportunities for improvement, currently, there exists no benchmark for scalable similarity search. A notable effort is [5]; however it covers only small in-memory datasets and a subset of the popular similarity search approaches. The community can build on this effort, leveraging a number of experimental evaluations conducted in this area [26, 27, 45].

3 RELATION TO PREVIOUS TUTORIALS

The similarity search problem is fundamental in computer science and has been addressed in previous tutorials [41, 61], which are over a decade old. The most recent relevant tutorial is [59]; however, its focus is on approximate techniques from the high-d community only, and does not cover a multitude of novel techniques with better scalability properties that, in addition, cover the entire spectrum of approximate to exact query answering. Our tutorial not only covers the state-of-the-art techniques in the field deriving from different communities, but also compares their performance, shares insights about their strengths and weaknesses, and emphasizes the key open research directions in the field.

Another relevant tutorial has appeared at ICDE [25]. That tutorial addresses single-node, exact and approximate similarity search techniques. In contrast, we focus on the following novel and emerging research directions in this area: (i) progressive exact and approximate similarity search; (ii) parallel and distributed exact and approximate similarity search; and (iii) AI-driven similarity search. These aspects are necessary for modern scalable and interactive data science applications handling massive high-d vector collections.

Detailed descriptions of the data series methods can be found in previous tutorials [22, 24].

4 PRESENTERS

Karima Echihabi is an Assistant Professor at UM6P, Morocco. She has conducted extensive experimental evaluations on high-d similarity search (published in PVLDB).

Kostas Zoumpatianos is a Marie Curie Fellow affiliated with Harvard Univ. and Univ. of Paris, working on data series management and adaptive data systems.

Themis Palpanas is Senior Member of the French Univ. Institute (IUF) and Professor at the Univ. of Paris, with expertise on data series management and analytics.

REFERENCES

[1] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. 2001. On the Surprising Behavior of Distance Metrics in High Dimensional Spaces. In *ICDT*.

[2] Abdullah Al-Mamun, Hao Wu, and Walid G. Aref. 2020. A Tutorial on Learned Multi-Dimensional Indexes *(SIGSPATIAL)*. 1–4.

[3] Uri Alon, Meital Zilberstein, Omer Levy, and Eran Yahav. 2019. Code2vec: Learning Distributed Representations of Code. 3, POPL (2019).

[4] Akhil Arora, Sakshi Sinha, Piyush Kumar, and Arnab Bhattacharya. 2018. HD-index: Pushing the Scalability-accuracy Boundary for Approximate kNN Search in High-dimensional Spaces. *PVLDB* 11, 8 (2018).

[5] Martin Aumüller, Erik Bernhardsson, and Alexander Faithfull. 2017. ANN-Benchmarks: A Benchmarking Tool for Approximate Nearest Neighbor Algorithms. In *SISAP*.

[6] A. Babenko and V. Lempitsky. 2015. The Inverted Multi-Index. *TPAMI* 37, 6 (2015).

[7] Bahman Bahmani, Ashish Goel, and Rajendra Shinde. 2012. Efficient Distributed Locality Sensitive Hashing *(CIKM)*.

[8] Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. 1990. The R*-tree: an efficient and robust access method for points and rectangles. In *SIGMOD*.

[9] Kevin S. Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. 1999. When Is "'Nearest Neighbor'" Meaningful?. In *ICDT*.

[10] Paul Boniol, John Paparrizos, Themis Palpanas, and Michael J. Franklin. 2021. SAND: Streaming Subsequence Anomaly Detection. *PVLDB* (2021).

[11] A. Broder. 1997. On the Resemblance and Containment of Documents. In *SEQUENCES*. 21–29.

[12] Sébastien Bubeck and Ulrike von Luxburg. 2009. Nearest Neighbor Clustering: A Baseline Method for Consistent Clustering with Arbitrary Objective Functions. *JMLR* 10 (2009).

[13] Simon Byers and Adrian E. Raftery. 1998. Nearest-Neighbor Clutter Removal for Estimating Features in Spatial Point Processes. *JASA* 93, 442 (1998).

[14] Deng Cai, Xiuye Gu, and Chaoqi Wang. 2017. A Revisit on Deep Hashings for Large-scale Content Based Image Retrieval. arXiv:1711.06016 [cs.CV]

[15] Paolo Ciaccia and Marco Patella. 2000. PAC Nearest Neighbor Queries: Approximate and Controlled Search in High-Dimensional and Metric Spaces.. In *ICDE*.

[16] Paolo Ciaccia, Marco Patella, and Pavel Zezula. 1997. M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In *VLDB*.

[17] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. 2004. Locality-sensitive Hashing Scheme Based on P-stable Distributions. In *SCG*.

[18] Wei Dong, Charikar Moses, and Kai Li. 2011. Efficient k-nearest neighbor graph construction for generic similarity measures. In *WWW*.

[19] Sumeet Dua and Xian Du. 2011. *Data Mining and Machine Learning in Cybersecurity* (1st ed.). Auerbach Publications, USA.

[20] Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq Joty, Mourad Ouzzani, and Nan Tang. 2018. Distributed Representations of Tuples for Entity Resolution. *VLDBJ* 11, 11 (2018).

[21] Karima Echihabi. 2020. High-Dimensional Similarity Search: From Time Series to Deep Network Embeddings. In *SIGMOD*.

[22] Karima Echihabi, Kostas Zoumpatianos, and Themis Palpanas. 2020. Big Sequence Management: on Scalability. In *IEEE BigData*.

[23] Karima Echihabi, Kostas Zoumpatianos, and Themis Palpanas. 2020. Scalable Machine Learning on High-Dimensional Vectors: From Data Series to Deep Network Embeddings. In *WIMS*.

[24] Karima Echihabi, Kostas Zoumpatianos, and Themis Palpanas. 2021. Big Sequence Management: Scaling up and Out. In *EDBT*.

[25] Karima Echihabi, Kostas Zoumpatianos, and Themis Palpanas. 2021. High-Dimensional Similarity Search for Scalable Data Science *(ICDE)*.

[26] Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, and Houda Benbrahim. 2018. The Lernaean Hydra of Data Series Similarity Search: An Experimental Evaluation of the State of the Art. *PVLDB* 12, 2 (2018).

[27] Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, and Houda Benbrahim. 2019. Return of the Lernaean Hydra: Experimental Evaluation of Data Series Approximate Similarity Search. *PVLDB* 13, 3 (2019).

[28] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. 1994. Fast subsequence matching in time-series databases. In *SIGMOD*.

[29] Hakan Ferhatosmanoglu, Ertem Tuncel, Divyakant Agrawal, and Amr El Abbadi. 2000. Vector Approximation Based Indexing for Non-uniform High Dimensional Data Sets *(CIKM)*.

[30] Cong Fu, Chao Xiang, Changxu Wang, and Deng Cai. 2019. Fast Approximate Nearest Neighbor Search with the Navigating Spreading-out Graph. *PVLDB*. 12, 5 (2019).

[31] Junhao Gan, Jianlin Feng, Qiong Fang, and Wilfred Ng. 2012. Locality-sensitive Hashing Scheme Based on Dynamic Collision Counting. In *SIGMOD*.

[32] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. 2014. Optimized Product Quantization. *TPAMI* 36, 4 (April 2014).

[33] Anna Gogolou, Theophanis Tsandilas, Karima Echihabi, Themis Palpanas, and Anastasia Bezerianos. 2020. Data Series Progressive Similarity Search with Probabilistic Quality Guarantees. In *SIGMOD*.

[34] Antonin Guttman. 1984. R-Trees: A Dynamic Index Structure for Spatial Searching. In *SIGMOD*.

[35] Junfeng He, Sanjiv Kumar, and Shih-Fu Chang. 2012. On the Difficulty of Nearest Neighbor Search. In *ICML*.

[36] Qiang Huang, Jianlin Feng, Yikai Zhang, Qiong Fang, and Wilfred Ng. 2015. Query-aware Locality-sensitive Hashing for Approximate Nearest Neighbor Search. *PVLDB* 9, 1 (2015), 1–12.

[37] Piotr Indyk and Rajeev Motwani. 1998. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality *(STOC)*.

[38] Suhas Jayaram Subramanya, Fnu Devvrit, Harsha Vardhan Simhadri, Ravishankar Krishnawamy, and Rohan Kadekodi. 2019. Rand-NSG: Fast Accurate Billion-point Nearest Neighbor Search on a Single Node. (2019).

[39] H. Jegou, M. Douze, and C. Schmid. 2011. Product Quantization for Nearest Neighbor Search. *TPAMI* 33, 1 (2011).

[40] Søren Kejser Jensen, Torben Bach Pedersen, and Christian Thomsen. 2017. Time Series Management Systems: A Survey. *TKDE* 29, 11 (2017).

[41] Jiawei Han and Xifeng Yan and Philip S. Yu. 2006. Mining, Indexing, and Similarity Search in Graphs and Complex Structures. In *ICDE*.

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

[43] Pauline Laviron, Xueqi Dai, Berenice Huquet, and Themis Palpanas. 2021. Electricity Demand Activation Extraction: From Known to Uknown Signatures, Using Similarity Search. In *e-Energy*.

[44] Conglong Li, Minjia Zhang, David G. Andersen, and Yuxiong He. 2020. Improving Approximate Nearest Neighbor Search through Learned Adaptive Early Termination. In *SIGMOD*. 2539–2554.

[45] W. Li, Y. Zhang, Y. Sun, W. Wang, M. Li, W. Zhang, and X. Lin. 2019. Approximate Nearest Neighbor Search on High Dimensional Data - Experiments, Analyses, and Improvement. *TKDE* (2019).

[46] Michele Linardi, Yan Zhu, Themis Palpanas, and Eamonn J. Keogh. 2020. Matrix Profile Goes MAD: Variable-Length Motif And Discord Discovery in Data Series. In *DAMI*.

[47] Ting Liu, Andrew W. Moore, Alexander Gray, and Ke Yang. 2004. An Investigation of Practical Approximate Nearest Neighbor Algorithms. In *NIPS*. 825–832.

[48] Xiao Luo, Chong Chen, Huasong Zhong, Hao Zhang, Minghua Deng, Jianqiang Huang, and Xiansheng Hua. 2020. A Survey on Deep Hashing Methods. arXiv:2003.03369 [cs.CV]

[49] 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).

[50] Stanislav Morozov and Artem Babenko. 2019. Unsupervised neural quantization for compressed-domain similarity search. In *ICCV*.

[51] Themis Palpanas. 2015. Data Series Management: The Road to Big Sequence Analytics. *SIGMOD Record* 44, 2 (2015).

[52] Themis Palpanas. 2020. Evolution of a Data Series Index: the iSAX Family of Data Series Indexes. *CCIS* 1197 (2020).

[53] Themis Palpanas and Volker Beckmann. 2019. Report on the First and Second Interdisciplinary Time Series Analysis Workshop (ITISA). *SIGMOD Rec.* 48, 3 (2019).

[54] Botao Peng, Panagiota Fatourou, and Themis Palpanas. 2021. Fast Data Series Indexing for In-Memory Data. *VLDBJ* (2021).

[55] Botao Peng, Panagiota Fatourou, and Themis Palpanas. 2021. SING: Sequence Indexing Using GPUs. In *ICDE*.

[56] Botao Peng, Themis Palpanas, and Panagiota Fatourou. 2020. ParIS+: Data Series Indexing on Multi-core Architectures. *TKDE* (2020).

[57] François Petitjean, Germain Forestier, Geoffrey I. Webb, Ann E. Nicholson, Yanping Chen, and Eamonn J. Keogh. 2014. Dynamic Time Warping Averaging of Time Series Allows Faster and More Accurate Classification. In *ICDM*.

[58] Liudmila Prokhorenkova and Aleksandr Shekhovtsov. 2020. Graph-based Nearest Neighbor Search: From Practice to Theory. In *PMLR*.

[59] Jianbin Qin, Wei Wang, Chuan Xiao, and Ying Zhang. 2020. Similarity Query Processing for High-Dimensional Data. *PVLDB* 13, 12 (2020).

[60] Hanan Samet. 2005. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann Publishers Inc.

[61] Hanan Samet. 2010. Techniques for Similarity Searching in Multimedia Databases. *PVLDB* 3, 2 (2010).

[62] S. Shekkizhar and A. Ortega. 2020. Graph Construction from Data by Non-Negative Kernel Regression. In *ICASSP*.

[63] Yifang Sun, Wei Wang, Jianbin Qin, Ying Zhang, and Xuemin Lin. 2014. SRS: Solving c-approximate Nearest Neighbor Queries in High Dimensional Euclidean Space with a Tiny Index. *PVLDB* 8, 1 (2014).

[64] Saravanan Thirumuruganathan, Shohedul Hasan, Nick Koudas, and Gautam Das. 2020. Approximate query processing for data exploration using deep generative models. In *ICDE*. 1309–1320.

[65] Cagatay Turkay, Erdem Kaya, Selim Balcisoy, and Helwig Hauser. 2017. Designing Progressive and Interactive Analytics Processes for High-Dimensional Data Analysis. *IEEE TVCG* 23, 1 (2017).

[66] Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. 2018. Billion-Scale Commodity Embedding for E-Commerce Recommendation in Alibaba. In *KDD*.

[67] J. Wang, T. Zhang, j. song, N. Sebe, and H. T. Shen. 2018. A Survey on Learning to Hash. *TPAMI* 40, 4 (2018).

[68] Qitong Wang and Themis Palpanas. 2021. Deep Learning Embeddings for Data Series Similarity Search. In *SIGKDD*.

[69] Yang Wang, Peng Wang, Jian Pei, Wei Wang, and Sheng Huang. 2013. A Data-adaptive and Dynamic Segmentation Index for Whole Matching on Time Series. *PVLDB* 6, 10 (2013).

[70] Roger Weber, Hans-Jörg Schek, and Stephen Blott. 1998. A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. In *Proc. VLDB*. 194–205.

[71] Hanwei Wu and Markus Flierl. 2019. Learning product codebooks using vector-quantized autoencoders for image retrieval. In *GlobalSIP*. IEEE, 1–5.

[72] Jiaye Wu, Peng Wang, Ningting Pan, Chen Wang, Wei Wang, and Jianmin Wang. 2019. KV-Match: A Subsequence Matching Approach Supporting Normalization and Time Warping. In *ICDE*.

[73] Djamel-Edine Yagoubi, Reza Akbarinia, Florent Masseglia, and Themis Palpanas. 2019. Massively Distributed Time Series Indexing and Querying. *TKDE* 32, 1 (2019).

[74] Emanuel Zgraggen, Alex Galakatos, Andrew Crotty, Jean-Daniel Fekete, and Tim Kraska. 2017. How Progressive Visualizations Affect Exploratory Analysis. *TVCG* 23, 8 (2017), 1977–1987.

[75] L. Zhang, N. Alghamdi, M. Y. Eltabakh, and E. A. Rundensteiner. 2019. TARDIS: Distributed Indexing Framework for Big Time Series Data. In *ICDE*.

[76] Erkang Zhu, Fatemeh Nargesian, Ken Q Pu, and Renée J Miller. 2016. LSH ensemble: internet-scale domain search. *PVLDB* 9, 12 (2016), 1185–1196.

[77] Kostas Zoumpatianos, Stratos Idreos, and Themis Palpanas. 2016. ADS: the adaptive data series index. *The VLDB Journal* 25, 6 (2016).