High-Dimensional Similarity Search for Scalable Data Science

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

Kostas Zoumpatianos
Harvard Univ.
kostas@seas.harvard.edu

Themis Palpanas
LIPADE, Univ. de Paris
themis@mi.parisdescartes.fr

***Abstract*—Similarity search is a core operation of many critical data science applications, involving massive collections of high-dimensional objects. Similarity search finds objects in a collection close to a given query according to some definition of sameness. 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-dimensional similarity search, we survey the state-of-the-art high-dimensional similarity search approaches and share surprising insights about their strengths and weaknesses, and we discuss the challenges and open research problems in this area.**

I. 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 [62]. In data integration, it has been used to automate entity resolution [29] and support data discovery [88]. It has powered recommender systems of online billion-dollar enterprises [78] and enabled clustering [14], classification [68] and outlier detection [11], [12], [15] 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 detect intrusions and malware [28].

This problem has been studied heavily in the past 25 years and will continue to attract attention as massive collections of high-dimensional objects are becoming omnipresent in various domains [63]. Objects can be data series, text, images, audio and video recordings, graphs, database tables or deep network embeddings. Similarity search over high-dimensional objects is often reduced to a k-Nearest Neighbor (k-NN) problem such that the objects are represented using high-dimensional vectors and the (dis)-similarity between them is measured using a distance. Some studies [1], [10] have argued that NN search is not meaningful for a number of high-dimensional 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-dimensional NN search is meaningful for non-i.i.d data, data with low intrinsic dimensionality and for a variety of real world datasets [39]. The importance and relevance of NN search in high-dimensions is further evidenced by a large and growing body of research [30], [31].

High-dimensional 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 do not provide any 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 the data science applications that require efficient high-dimensional similarity search, provides an overview of the state-of-the-art exact and approximate high-dimensional similarity search approaches and discusses the challenges and open research problems in this domain.

II. Data Science Applications

Data integration. Similarity search is used by data integration tools to facilitate a variety of tasks. Among others, it is used over tuple embeddings for entity resolution [29], and over schema embeddings for data discovery [59].

Recommender Systems. Similarity search is used in recommender systems to predict the interest of a user for a new item. Item embeddings are learned from user-item interactions, and used to recommend movies [27] and products [78].

Information Retrieval. Similarity search is used in information retrieval for finding similar multimedia objects and copy detection in document [58], and video [72] collections.

Software Engineering. Code embeddings have been proposed to represent arbitrary snippets of code [3], and can be used to predict software dependencies and code clones.

Cybersecurity. Similarity search has supported critical cybersecurity operations, such as profiling network usage, and detecting intrusions and malware [28].

Outlier Detection. Techniques based on similarity search define outliers as the items with the largest distance to their

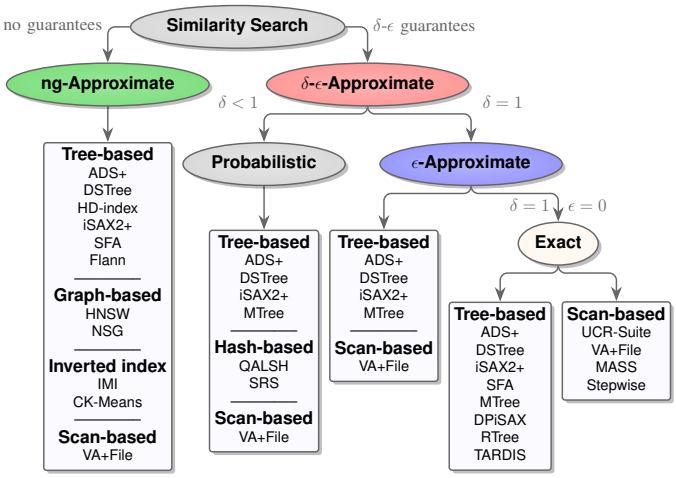

Fig. 1. Taxonomy of similarity search methods.

NNs, and have been applied to find land mines in satellite images [15] and discords in data series [54].

Classification. Similarity-based classification (e.g., k-NN Classifier) is used in a variety of domains, such as bioinformatics for protein classification [4], and remote sensing [68].

Clustering Similarity search has been used for clustering by finding the k closest neighbors around a given point [14].

III. State of the Art and Future Opportunities

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 [32], [33], in order to facilitate further work in this area.

Exact techniques guarantee correct results at the expense of efficiency and footprint. The research community has developed exact approaches for generic high-dimensional vectors [9], [23], [34], [81] (for an exhaustive survey, see [73]) and specific ones for data series [2], [16], [17], [19]–[21], [46], [48]–[50], [52], [53], [60], [64]–[67], [70], [71], [75], [76], [80], [82], [84]–[86], [89]. We discuss, contrast and compare these techniques.

We also 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 and fast decision making [38].

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 similarity search is making the right trade-offs between accuracy, efficiency and footprint. We note that several of the data series techniques for exact query answering that we mentioned above, also support (the different flavors of) approximate search, which we discuss below.

Approximate Search With Guarantees. δ-ϵ-approximate search dates from 1998 [41] and gave rise to a rich family of LSH algorithms [79], which solve the problem in sub-linear time, for $\delta < 1$. The main idea is that two neighbors in a high dimensional 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 [13], [18], [24], [36], or improving the theoretical bounds on query accuracy (i.e., δ or ϵ), query efficiency or the index size [40], [55], [77].

A δ-ϵ-approximate search algorithm was also proposed for the MTree [22], and the same ideas were used to extend existing exact data series techniques to enable them to support δ-ϵ-approximate search [33]. These extensions surprisingly outperformed the MTree and the state-of-the-art LSH techniques [40], [77] across the board in efficiency, accuracy and footprint, in-memory and on-disk.

Promising directions include developing scalable LSH techniques [87], exploiting deep learning for data-dependent hashing [56], and devising effective stopping criteria to further improve the efficiency of data series extensions [33].

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 [26], [35], [57] and inverted indexes [7], [37], [43], [83]. HNSW [7], [57], a proximity method based on navigable small world graphs, is considered the best contender for in-memory ng-approximate search [6], [33], [51], while data series similarity search methods have superior performance on-disk [33].

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 [42].

A. 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 [63].

Extending Guarantees. 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) [55]. 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 [33]. 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. Recall and MAP are better indicators of accuracy, because even small approximation errors may still result in low recall/MAP values [5], [33].

B. Other Considerations

While most studies have focused on the high-dimensional 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-dimensional 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 [44], though, more advanced and general systems are needed [63].

C. Benchmarks

Benchmarking is important because it allows a fair comparison of different solutions, helps foster reproducible research and can serve to identify gaps in the state-of-the-art, which would in turn spur future research developments. 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 [6], which proposes an interactive on-line benchmarking environment for approximate NN search; however it covers only small in-memory datasets and a subset of the popular similarity search approaches. A recent work studies the concept of hardness of NN queries, and proposes a method that constructs synthetically harder workloads [90]. The community can expand on these efforts, leveraging a number of experimental evaluations conducted in this area. Some studies focus on the accuracy of similarity measures and dimensionality reduction techniques [8], [25], [47], while others on the efficiency of exact methods [32], or the efficiency and accuracy of approximate approaches [6], [33], [51], [61]. We will share the key insights gained from these studies, describing the strengths and weaknesses of the various approaches and linking their performance behavior to their design choices.

IV. Scope and Outline

This is a 3-hour tutorial. It will motivate high-dimensional similarity search in the context of data science in 15min (§2), and will mainly focus on the key problems in the field, covering the state-of-the-art and open research directions for each problem (§3). This tutorial is designed for data science researchers and practitioners, and will include the necessary background for newcomers.

V. Relation to Previous Tutorials

The similarity search problem is fundamental in computer science and has been addressed in previous tutorials [45], [74], which are over a decade old. The most recent relevant tutorial is [69]; however, its focus is on approximate techniques from the high-dimensional 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.

VI. Presenters

Karima Echihabi is an Assistant Professor at Mohammed VI University in Morocco. Her research interests lie in scalable data analytics, and has conducted the two most extensive experimental evaluations in the area of high-dimensional similarity search (published in PVLDB). She has worked in the Windows team at Microsoft Redmond, and the Query Optimizer team at the IBM Toronto Lab.

Kostas Zoumpatianos has been a Marie Curie Fellow affiliated with Harvard Univ. and Univ. of Paris. He holds a PhD from the Univ. of Trento. He has visited Cornell Univ. His research is in the domains of data series management, analytics mining, self-designing, and adaptive data systems. He has developed the ADS and Coconut data series indexes, and has published in top international database venues.

Themis Palpanas is a Senior Member of the French Univ. Institute (IUF), and a professor at the Univ. of Paris. He is the author of 9 US patents. His focus in on Data Series Management and Analytics, developing and publishing several of the state of the art techniques in major journals and conferences. He has delivered 8 tutorials in top international conferences, including VLDB and SIGMOD, ICDE, and EDBT.

References

[1] C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional spaces. In *ICDT*, 2001.

[2] R. Agrawal, C. Faloutsos, and A. N. Swami. Efficient similarity search in sequence databases. In *FODO*, 1993.

[3] U. Alon, M. Zilberstein, O. Levy, and E. Yahav. Code2vec: Learning distributed representations of code. 3(POPL), 2019.

[4] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl. Nearest neighbor classification in 3d protein databases. *ISMB*, 1999.

[5] A. Arora, S. Sinha, P. Kumar, and A. Bhattacharya. HD-index: Pushing the Scalability-accuracy Boundary for Approximate kNN Search in High-dimensional Spaces. *PVLDB*, 11(8), 2018.

[6] M. Aumüller, E. Bernhardsson, and A. Faithfull. ANN-Benchmarks: A Benchmarking Tool for Approximate Nearest Neighbor Algorithms. In *SISAP*, 2017.

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

[8] A. J. Bagnall, J. Lines, A. Bostrom, J. Large, and E. J. Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *DMKD*, 31(3), 2017.

[9] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: an efficient and robust access method for points and rectangles. In *SIGMOD*, 1990.

[10] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "'nearest neighbor'" meaningful? In *ICDT*, 1999.

[11] P. Boniol, M. Linardi, F. Roncallo, and T. Palpanas. Automated anomaly detection in large sequences. In *ICDE*, 2020.

[12] P. Boniol and T. Palpanas. Series2graph: Graph-based subsequence anomaly detection for time series. *PVLDB*, 13(11), 2020.

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

[14] S. Bubeck and U. von Luxburg. Nearest neighbor clustering: A baseline method for consistent clustering with arbitrary objective functions. *JMLR*, 10, 2009.

[15] S. Byers and A. E. Raftery. Nearest-neighbor clutter removal for estimating features in spatial point processes. *JASA*, 93(442), 1998.

[16] A. Camerra, T. Palpanas, J. Shieh, and E. J. Keogh. iSAX 2.0: Indexing and Mining One Billion Time Series. In *ICDM*, 2010.

[17] A. Camerra, J. Shieh, T. Palpanas, T. Rakthanmanon, and E. J. Keogh. Beyond one billion time series: indexing and mining very large time series collections with iSAX2+. *KAIS*, 39(1), 2014.

[18] M. S. Charikar. Similarity Estimation Techniques from Rounding Algorithms. In *STOC*, 2002.

[19] G. Chatzigeorgakidis, D. Skoutas, K. Patroumpas, T. Palpanas, S. Athanasiou, and S. Skiadopoulos. Local pair and bundle discovery over co-evolving time series. In *SSTD*, 2019.

[20] G. Chatzigeorgakidis, D. Skoutas, K. Patroumpas, T. Palpanas, S. Athanasiou, and S. Skiadopoulos. Local similarity search on geolocated time series using hybrid indexing. In *SIGSPATIAL*, 2019.

[21] G. Chatzigeorgakidis, D. Skoutas, K. Patroumpas, T. Palpanas, S. Athanasiou, and S. Skiadopoulos. Twin subsequence search in time series. In *EDBT*, 2021.

[22] P. Ciaccia and M. Patella. PAC Nearest Neighbor Queries: Approximate and Controlled Search in High-Dimensional and Metric Spaces. In *ICDE*, 2000.

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

[24] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive Hashing Scheme Based on P-stable Distributions. In *SCG*, 2004.

[25] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *VLDB*, 2008.

[26] W. Dong, C. Moses, and K. Li. Efficient k-nearest neighbor graph construction for generic similarity measures. In *Proceedings of the 20th international conference on World wide web*, pages 577–586, 2011.

[27] X. Dong, L. Yu, Z. Wu, Y. Sun, L. Yuan, and F. Zhang. A hybrid collaborative filtering model with deep structure for recommender systems. In *AAAI*, 2017.

[28] S. Dua and X. Du. *Data Mining and Machine Learning in Cybersecurity*. Auerbach Publications, USA, 1st edition, 2011.

[29] M. Ebraheem, S. Thirumuruganathan, S. Joty, M. Ouzzani, and N. Tang. Distributed representations of tuples for entity resolution. *VLDBJ*, 11(11), 2018.

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

[31] K. Echihabi, K. Zoumpatianos, and T. Palpanas. Scalable machine learning on high-dimensional vectors: From data series to deep network embeddings. In *WIMS*, 2020.

[32] K. Echihabi, K. Zoumpatianos, T. Palpanas, and H. Benbrahim. The Lernaean Hydra of Data Series Similarity Search: An Experimental Evaluation of the State of the Art. *PVLDB*, 12(2), 2018.

[33] K. Echihabi, K. Zoumpatianos, T. Palpanas, and H. Benbrahim. Return of the Lernaean Hydra: Experimental Evaluation of Data Series Approximate Similarity Search. *PVLDB*, 13(3), 2019.

[34] H. Ferhatosmanoglu, E. Tuncel, D. Agrawal, and A. El Abbadi. Vector Approximation Based Indexing for Non-uniform High Dimensional Data Sets. CIKM, 2000.

[35] C. Fu, C. Xiang, C. Wang, and D. Cai. Fast Approximate Nearest Neighbor Search with the Navigating Spreading-out Graph. *PVLDB.*, 12(5), 2019.

[36] J. Gan, J. Feng, Q. Fang, and W. Ng. Locality-sensitive Hashing Scheme Based on Dynamic Collision Counting. In *SIGMOD*, 2012.

[37] T. Ge, K. He, Q. Ke, and J. Sun. Optimized Product Quantization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(4):744–755, Apr. 2014.

[38] A. Gogolou, T. Tsandilas, K. Echihabi, T. Palpanas, and A. Bezerianos. Data Series Progressive Similarity Search with Probabilistic Quality Guarantees. In *SIGMOD*, 2020.

[39] J. He, S. Kumar, and S.-F. Chang. On the difficulty of nearest neighbor search. In *ICML*, 2012.

[40] Q. Huang, J. Feng, Y. Zhang, Q. Fang, and W. Ng. Query-aware Locality-sensitive Hashing for Approximate Nearest Neighbor Search. *PVLDB*, 9(1):1–12, 2015.

[41] P. Indyk and R. Motwani. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. STOC, 1998.

[42] S. Jayaram Subramanya, F. Devvrit, H. V. Simhadri, R. Krishnawamy, and R. Kadekodi. Diskann: Fast accurate billion-point nearest neighbor search on a single node. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

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

[44] S. K. Jensen, T. B. Pedersen, and C. Thomsen. Time series management systems: A survey. *TKDE*, 29(11), 2017.

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

[46] S. Kashyap and P. Karras. Scalable kNN search on vertically stored time series. In *KDD*, 2011.

[47] E. Keogh and S. Kasetty. On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. *DMKD*, 7(4), 2003.

[48] H. Kondylakis, N. Dayan, K. Zoumpatianos, and T. Palpanas. Coconut: A Scalable Bottom-Up Approach for Building Data Series Indexes. *PVLDB*, 11(6), 2018.

[49] H. Kondylakis, N. Dayan, K. Zoumpatianos, and T. Palpanas. Coconut: Sortable summarizations for scalable indexes over static and streaming data series. *VLDBJ*, 28(6), 2019.

[50] O. Levchenko, B. Kolev, D. E. Yagoubi, R. Akbarinia, F. Masseglia, T. Palpanas, D. E. Shasha, and P. Valduriez. Bestneighbor: efficient evaluation of knn queries on large time series databases. *KAIS*, 63(2), 2021.

[51] 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. *TKDE*, 2019.

[52] M. Linardi and T. Palpanas. Scalable, variable-length similarity search in data series: The ulisse approach. *PVLDB*, 11(13), 2018.

[53] M. Linardi and T. Palpanas. ULISSE: ULtra compact Index for Variable-Length Similarity SEarch in Data Series. In *ICDE*, 2018.

[54] M. Linardi, Y. Zhu, T. Palpanas, and E. J. Keogh. Matrix Profile Goes MAD: Variable-Length Motif And Discord Discovery in Data Series. In *DAMI*, 2020.

[55] T. Liu, A. W. Moore, A. Gray, and K. Yang. An Investigation of Practical Approximate Nearest Neighbor Algorithms. In *NIPS*, pages 825–832, 2004.

[56] X. Luo, C. Chen, H. Zhong, H. Zhang, M. Deng, J. Huang, and X. Hua. A survey on deep hashing methods, 2020.

[57] Y. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. *CoRR*, abs/1603.09320, 2016.

[58] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. NIPS, 2013.

[59] R. J. Miller. Open data integration. *PVLDB*, 11(12), 2018.

[60] A. Mueen, Y. Zhu, M. Yeh, K. Kamgar, K. Viswanathan, C. Gupta, and E. Keogh. The Fastest Similarity Search Algorithm for Time Series Subsequences under Euclidean Distance, 2017.

[61] B. Naidan, L. Boytsov, and E. Nyberg. Permutation Search Methods Are Efficient, Yet Faster Search is Possible. *PVLDB*, 8(12), 2015.

[62] T. Palpanas. Data series management: The road to big sequence analytics. *SIGMOD Record*, 2015.

[63] T. Palpanas and V. Beckmann. Report on the first and second interdisciplinary time series analysis workshop (itisa). *SIGMOD Rec.*, 48(3), 2019.

[64] B. Peng, P. Fatourou, and T. Palpanas. ParIS: The Next Destination for Fast Data Series Indexing and Query Answering. *IEEE BigData*, 2018.

[65] B. Peng, P. Fatourou, and T. Palpanas. MESSI: In-Memory Data Series Indexing. *ICDE*, 2020.

[66] B. Peng, T. Palpanas, and P. Fatourou. Paris+: Data series indexing on multi-core architectures. *TKDE*, 2020.

[67] B. Peng, T. Palpanas, and P. Fatourou. SING: Sequence Indexing Using GPUs. In *ICDE*, 2021.

[68] F. Petitjean, G. Forestier, G. I. Webb, A. E. Nicholson, Y. Chen, and E. J. Keogh. Dynamic time warping averaging of time series allows faster and more accurate classification. In *ICDM*, 2014.

[69] J. Qin, W. Wang, C. Xiao, and Y. Zhang. Similarity query processing for high-dimensional data. *PVLDB*, 13(12), 2020.

[70] D. Rafiei. On Similarity-Based Queries for Time Series Data. In *ICDE*, 1999.

[71] T. Rakthanmanon, B. J. L. Campana, A. Mueen, G. E. A. P. A. Batista, M. B. Westover, Q. Zhu, J. Zakaria, and E. J. Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In *KDD*, 2012.

[72] V. Ramanathan, K. Tang, G. Mori, and L. Fei-Fei. Learning temporal embeddings for complex video analysis. In *ICCV*, 2015.

[73] H. Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann Publishers Inc., 2005.

[74] H. Samet. Techniques for similarity searching in multimedia databases. *PVLDB*, 3(2), 2010.

[75] P. Schäfer and M. Högqvist. SFA: A Symbolic Fourier Approximation and Index for Similarity Search in High Dimensional Datasets. EDBT, 2012.

[76] J. Shieh and E. Keogh. iSAX: Indexing and Mining Terabyte Sized Time Series. In *KDD*, pages 623–631, New York, NY, USA, 2008. ACM.

[77] 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), 2014.

[78] J. Wang, P. Huang, H. Zhao, Z. Zhang, B. Zhao, and D. L. Lee. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In *KDD*, 2018.

[79] J. Wang, T. Zhang, j. song, N. Sebe, and H. T. Shen. A survey on learning to hash. *TPAMI*, 40(4), 2018.

[80] Y. Wang, P. Wang, J. Pei, W. Wang, and S. Huang. A Data-adaptive and Dynamic Segmentation Index for Whole Matching on Time Series. *PVLDB*, 6(10), 2013.

[81] R. Weber, H.-J. Schek, and S. Blott. A Quantitative Analysis and Performance Study for Similarity-Search Methods in High-Dimensional Spaces. In *Proc. VLDB*, pages 194–205, 1998.

[82] J. Wu, P. Wang, N. Pan, C. Wang, W. Wang, and J. Wang. Kv-match: A subsequence matching approach supporting normalization and time warping. In *ICDE*, 2019.

[83] Y. Xia, K. He, F. Wen, and J. Sun. Joint Inverted Indexing. *ICCV*, 2013.

[84] D. E. Yagoubi, R. Akbarinia, F. Masseglia, and T. Palpanas. DPiSAX: Massively Distributed Partitioned iSAX. In *ICDM*, 2017.

[85] D.-E. Yagoubi, R. Akbarinia, F. Masseglia, and T. Palpanas. Massively distributed time series indexing and querying. *TKDE*, 32(1), 2019.

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

[87] 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.

[88] E. Zhu, F. Nargesian, K. Q. Pu, and R. J. Miller. Lsh ensemble: internet-scale domain search. *Proceedings of the VLDB Endowment*, 9(12):1185–1196, 2016.

[89] K. Zoumpatianos, S. Idreos, and T. Palpanas. ADS: the adaptive data series index. *The VLDB Journal*, 25(6), 2016.

[90] K. Zoumpatianos, Y. Lou, I. Ileana, T. Palpanas, and J. Gehrke. Generating data series query workloads. *VLDBJ*, 27(6), 2018.