Big Sequence Management

Kostas Zoumpatianos
University of Trento
zoumpatianos@disi.unitn.it

Michele Linardi
Paris Descartes University
michele.linardi@parisdescartes.fr

Themis Palpanas
Paris Descartes University
themis@mi.parisdescartes.fr

1. EXTENDED ABSTRACT

[Motivation.] Data series have gathered the attention of the data management community for almost two decades [27, 6, 18]. Data series are one of the most common types of data, and are present in virtually every scientific and social domain: they appear as audio sequences [12], shape and image data [31], financial [26], environmental monitoring [24] and scientific data [11], and they have many diverse applications, such as in health care, astronomy, biology, economics, and others.

Recent advances in sensing, networking, data processing and storage technologies have significantly eased the process of generating and collecting tremendous amounts of data series at extremely high rates and volumes. It is not unusual for applications to involve numbers of sequences in the order of hundreds of millions to billions [1, 2].

[Data Series.] A *data series*, or *data sequence*, is an ordered sequence of data points[1]. Formally, a data series $T = (p_1, ...p_n)$ is defined as a sequence of points $p_i = (v_i, t_i)$, where each point is associated with a value v_i and a time t_i in which this recording was made, and n is the size (or length) of the series. If the dimension that imposes the ordering of the sequence is time then we talk about *time series*, though, a series can also be defined over other measures (e.g., angle in radial profiles in astronomy, mass in mass spectroscopy, position in genome sequences, etc.).

A key observation is that analysts need to process and analyze a sequence (or subsequence) of values as a single object, rather than the individual points independently, which is what makes the management and analysis of data sequences a hard problem. Note that even though a sequence can be regarded as a point in n-dimensional space, traditional multi-dimensional approaches fail in this case, mainly due to the combination of the following two reasons: (a) the dimensionality is typically very high, i.e., in the order of several hundreds to several thousands, and (b) dimensions are strictly ordered (imposed by the sequence itself) and neighboring values are correlated.

[Need for Data Series Indexing.] In this context, nearest neighbor queries are of paramount importance, since they form the basis of virtually every data mining, or other complex analysis task involving data series. However, nearest neighbor queries across a large collection of data series are challenging, because data series collections grow very large in practice, with datasets including billions, or even trillions of data series [7, 23]. Thus, methods for answering nearest neighbor queries rely on two main techniques: data summarization and indexing. Data series summarization is used to reduce the dimensionality of the data series [14, 22, 16, 3, 13, 8, 17], and then indexes are built on top of these summarizations [22, 27, 4, 25, 30].

Nevertheless, as the data series collections grow in size, the operation of indexing these collections can itself become the bottleneck in the entire process. As an answer to this problem, we have developed the iSAX2.0 [6] and iSAX2+ [7], the first data series indexes that inherently support bulk loading, and thus aim to minimize the index building time. Bulk loading refers to mechanisms that allow us to insert at once a large quantity of data in an index, and as a result lead to fast index-building times. Furthermore, we describe the ADS+ index [32, 33], which is the first data series index than can start answering queries correctly before the entire index has been built. This goal is achieved by building very fast the main-memory part of the index (i.e., only the inner nodes), and deferring the materialization of the (expensive) leaf nodes to query time. This novel approach considerably shrinks the data-to-query gap, allowing users to start answering queries much faster than any previous approach, and enabling truly exploratory analysis on very large data series collections.

[Need for Data Series Management Systems.] There are important reasons why data Series (or Sequence) Management Systems (SMSs) are on the cusp of becoming a focal point for research activity in data management. The solutions that are currently available require custom code and the development of ad hoc systems for various tasks, requiring huge investments in time and effort, and duplication of effort across different teams. Even existing approaches based on DBMSs [5], Column Stores [28], or Array Databases [29]) do not provide a viable solution, since they have not been designed for managing and processing sequence data. Therefore, they do not offer a suitable declarative query language, storage model, auxiliary data structures (such as indexes), and optimization mechanism that can support a variety of

[1]For the rest of this paper, we are going to use the terms *data series* and *sequence* interchangeably.

sequence query workloads in an efficient manner.

We argue that a SMS is necessary in order to enable big sequence analytics, since it will offer the abstractions, tools, and automations needed for achieving this goal. Just like databases abstracted the relational data management problem and offered a black box solution that is now omnipresent, the proposed system will make it feasible for analysts that are not experts in data series management, as well as common users, to tap in the goldmine of the massive and ever-growing data series collections they (already) have.

[Contributions.] We briefly review the work relevant to data series summarization, and data series indexing. We present in more detail the iSAX summarization method, and discuss how it can be used to construct a data series index. Furthermore, we give an overview of the first data series indexes that support bulk loading, namely, iSAX2.0 and iSAX2+, which lead to index-building times considerably faster than previous approaches, allowing us to index datasets with 1 billion data series.

We describe the first adaptive data series index, ADS+, which reduces by an additional order of magnitude the time needed by the index before it is ready to start answering queries. The ADS+ index starts by a minimal tree structure based on summarizations of the data series. Then, the index structure is continuously enriched as more queries arrive: each query that is not covered by the current contents of the index, triggers additional data to be brought inside the index, thus adaptively and automatically expanding subtrees in the hot branches of the index. This enables ADS+ to answer several hundreds of thousands of queries by the time that state-of-the-art techniques are still in the index creation phase.

We argue for the need to develop a general-purpose sequence management system, and discuss the features of such a system: (a) it should be able to cope with big data sequences, that is, massive collections of sequences, which can be heterogeneous (i.e., originate from disparate domains and thus exhibit very different characteristics), and which can have uncertainty in their values (e.g., due to inherent errors in the measurements); (b) it should efficiently support a wide range of sequence queries and mining operations at a scalable fashion, while exploiting the benefits of physical and logical independence; and (c) it should support cost-based optimization, which will enable the system to automatically pick the right storage and execution strategies for answering different queries.

References

[1] Adhd-200. `http://fcon_1000.projects.nitrc.org/indi/adhd200/`, 2011.

[2] Sloan digital sky survey. `https://www.sdss3.org/dr10/data_access/volume.php`, 2015.

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

[4] I. Assent, R. Krieger, F. Afschari, and T. Seidl. The ts-tree: Efficient time series search and retrieval. In *EDBT*, 2008.

[5] M. M. Astrahan, M. W. Blasgen, D. D. Chamberlin, K. P. Eswaran, J. Gray, P. P. Griffiths, W. F. K. III, R. A. Lorie, P. R. McJones, J. W. Mehl, G. R. Putzolu, I. L. Traiger, B. W. Wade, and V. Watson. System R: relational approach to database management. *TODS*, 1(2):97–137, 1976.

[6] A. Camerra, T. Palpanas, J. Shieh, and E. Keogh. iSAX 2.0: Indexing and mining one billion time series. In *ICDM*, 2010.

[7] 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):123–151, 2014.

[8] K.-P. Chan and A.-C. Fu. Efficient time series matching by wavelets. In *ICDE*, 1999.

[9] M. Dallachiesa, B. Nushi, K. Mirylenka, and T. Palpanas. Uncertain time-series similarity: Return to the basics. *PVLDB*, 5(11):1662–1673, 2012.

[10] M. Dallachiesa, T. Palpanas, and I. F. Ilyas. Top-k nearest neighbor search in uncertain data series. *PVLDB*, 8(1):13–24, 2014.

[11] P. Huijse, P. A. Estévez, P. Protopapas, J. C. Principe, and P. Zegers. Computational intelligence challenges and applications on large-scale astronomical time series databases. *IEEE Comp. Int. Mag.*, 9(3):27–39, 2014.

[12] K. Kashino, G. Smith, and H. Murase. Time-series active search for quick retrieval of audio and video. In *ICASSP*, 1999.

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

[14] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *KAIS*, 3(3):263–286, 2000.

[15] E. J. Keogh, T. Palpanas, V. B. Zordan, D. Gunopulos, and M. Cardle. Indexing large human-motion databases. In *VLDB*, pages 780–791, 2004.

[16] C.-S. Li, P. Yu, and V. Castelli. Hierarchyscan: a hierarchical similarity search algorithm for databases of long sequences. In *ICDE*, 1996.

[17] J. Lin, E. Keogh, S. Lonardi, and B. Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *DMKD*, 2003.

[18] J. Lin, R. Khade, and Y. Li. Rotation-invariant similarity in time series using bag-of-patterns representation. *J. Intell. Inf. Syst.*, 39(2), 2012.

[19] T. Palpanas. Data series management: The road to big sequence analytics. *SIGMOD Rec.*, 44(2):47–52, 2015.

[20] T. Palpanas, M. Vlachos, E. J. Keogh, and D. Gunopulos. Streaming time series summarization using user-defined amnesic functions. *IEEE Trans. Knowl. Data Eng.*, 20(7):992–1006, 2008.

[21] T. Palpanas, M. Vlachos, E. J. Keogh, D. Gunopulos, and W. Truppel. Online amnesic approximation of streaming time series. In *ICDE*, pages 339–349, 2004.

[22] D. Rafiei and A. Mendelzon. Similarity-based queries for time series data. In *SIGMOD*, 1997.

[23] T. Rakthanmanon, B. J. L. Campana, A. Mueen, G. 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.

[24] U. Raza, A. Camerra, A. L. Murphy, T. Palpanas, and G. P. Picco. Practical data prediction for real-world wireless sensor networks. *IEEE Trans. Knowl. Data Eng.*, accepted for publication, 2015.

[25] P. Schäfer and M. Högqvist. Sfa: A symbolic fourier approximation and index for similarity search in high dimensional datasets. In *EDBT*, 2012.

[26] D. Shasha. Tuning time series queries in finance: Case studies and recommendations. *IEEE Data Eng. Bull.*, 22(2):40–46, 1999.

[27] J. Shieh and E. J. Keogh. *i*sax: indexing and mining terabyte sized time series. In *KDD*, pages 623–631, 2008.

[28] M. Stonebraker, D. J. Abadi, A. Batkin, X. Chen, M. Cherniack, M. Ferreira, E. Lau, A. Lin, S. Madden, E. J. O'Neil, P. E. O'Neil, A. Rasin, N. Tran, and S. B. Zdonik. C-store: A column-oriented DBMS. In *VLDB*, 2005.

[29] M. Stonebraker, P. Brown, A. Poliakov, and S. Raman. The architecture of scidb. In *SSDBM*, 2011.

[30] 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):793–804, 2013.

[31] L. Ye and E. J. Keogh. Time series shapelets: a new primitive for data mining. In *KDD*, 2009.

[32] K. Zoumpatianos, S. Idreos, and T. Palpanas. Indexing for interactive exploration of big data series. In *SIGMOD*, 2014.

[33] K. Zoumpatianos, S. Idreos, and T. Palpanas. RINSE: interactive data series exploration with ADS+. *PVLDB*, 8(12):1912–1923, 2015.

[34] K. Zoumpatianos, Y. Lou, T. Palpanas, and J. Gehrke. Query workloads for data series indexes. In *KDD*, 2015.