Sim-Piece+: Efficient Time Series Data Compression

Xenophon Kitsios
xkitsios@aueb.gr
Athens University of Economics and Business
Athens, Greece

Panagiotis Liakos
panagiotisliakos@aueb.gr
Athens University of Economics and Business
Athens, Greece

Katia Papakonstantinopoulou
katia@aueb.gr
Athens University of Economics and Business
Athens, Greece

Yannis Kotidis
kotidis@aueb.gr
Athens University of Economics and Business
Athens, Greece

ABSTRACT

Lossy compression enhances the scalability of contemporary data collection infrastructures by reducing the volume of data that needs to be transmitted and stored. Sim-Piece is a state-of-the-art lossy compressing algorithm for time series data that outperforms competing techniques, attaining compression ratios with more than twofold improvement on average over what prior lossy algorithms can offer. In this work, we introduce Sim-Piece+, an enhanced iteration of Sim-Piece that further enhances compression ratios through the utilization of advanced data encoding techniques.

ACM Reference Format:
Xenophon Kitsios, Panagiotis Liakos, Katia Papakonstantinopoulou, and Yannis Kotidis. 2023. Sim-Piece+: Efficient Time Series Data Compression. In *Proceedings of 5th Summit on Gender Equality in Computing (GEC '23).* ACM, New York, NY, USA, 2 pages.

1 INTRODUCTION

With the emergence of the Internet of Things (IoT), enormous amounts of streaming, timestamped datasets are being generated. Sensors from smart wearables, smart cities, autonomous cars, agricultural facilities etc., produce time series data which needs to be collected centrally in order to be later analyzed [1, 2, 3, 5, 6, 8]. Time series are also encountered in other domains, such as finance, e-commerce, health care, social networks and more.

Specialized time series data compression algorithms, such as Gorilla [14] and Chimp [11], are utilized to enable Time Series Databases (TSDBs) to effectively handle extensive datasets. These algorithms are designed for applications that require precise and detailed data values. On the other hand, lossy compression algorithms provide significantly greater space savings by allowing a certain level of acceptable error within a predefined bound. These algorithms are particularly suitable for applications that put emphasis on extracting meaningful patterns and insights from the data, rather than preserving precise numerical accuracy. They find utility in tasks such as seasonality detection, clustering, forecasting, and similarity search.

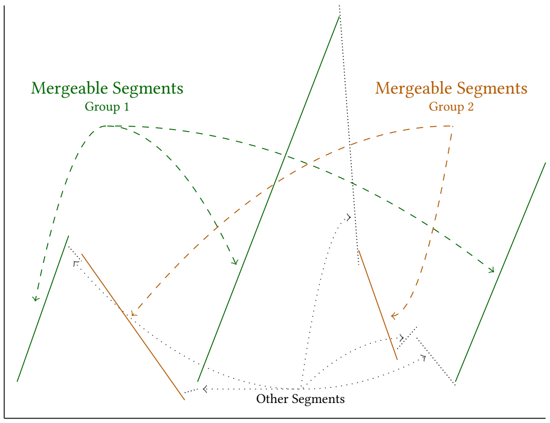

Figure 1: Groups of segments that we can represent jointly with Sim-Piece.

Sim-Piece [9] is a lossy timeseries algorithm that is based on the popular Piecewise Linear Approximation (PLA) scheme [7]. The novel algorithm takes advantage of the resemblances among the line segments generated by the PLA algorithm to create a unified representation for multiple segments. Our experimental evaluation using multiple real and synthetic datasets [9] demonstrates significant improvements in space efficiency across a wide range of maximum error values, even when the permissible error margin is minimal. In this work we present Sim-Piece+, an extension of Sim-Piece that offers even more substantial reductions in space requirements by improving the encoding of its internal representation.

2 SIM-PIECE ALGORITHM

PLA algorithms represent time-series measurements using a sequence of line segments, while keeping the approximation error within a predetermined acceptable threshold. Sim-Piece is a novel approach for PLA that seeks to exploit similarities among the line segments produced. Sim-Piece starts a line segment at a quantized value close to the original one. Then, it adds more points by keeping two slopes: the highest and lowest ones that fit the data within the required approximation guarantees, until a point goes beyond them. As any of the lines between the two slopes can approximate the data points of the segment, we can find groups of segments with intersecting sets of candidate lines and represent them jointly, to reduce the overall space requirements. Figure 1 shows an example

GEC '23, June 27, 2023, Athens, GR

Variable	Sim-Piece	Sim-Piece+
$error$	FLOAT	FLOAT
b_1	INT	INT
$\#b$	UINT	VARIABLE-BYTE
for $i = 1; i \leq \#b; i++;$		
b_i	BYTE (+UINT)*	BYTE (+UINT)*
$\#a_i$	BYTE (+SHORT)	VARIABLE-BYTE
for $j = 1; j \leq \#a_i; j++$		
$a_{i,j}$	FLOAT	FLOAT
$\#t_{i,j}$	BYTE (+SHORT)	VARIABLE-BYTE*
for $k = 1; k \leq \#t_{i,j}; k++$		
$t_{i,j,k}$	UINT	VARIABLE-BYTE
t_{last}	UINT	VARIABLE-BYTE

* Delta Encoding

Table 1: Data representation for encoding Sim-Piece data.

PLA approximation with two groups of mergable segments. Each group can be represented with a single pair of (starting-value, slope) values. This allows us to represent, in this example, all five PLA segments with using just two lines. Their starting timestamps are also stored, so that we can reconstruct them entirely when decompressing them. Given a set of PLA segments, Sim-Piece computes the optimal solution to this problem, coming up with the minimum possible number of groups to represent them jointly [9].

The encoding of Sim-Piece data is presented in Table 1. Each segment is represented by a line equation $y_i(t) = a_i(t - t_i) + b_i$, where a_i denotes the slope, b_i represents the starting value, and t_i signifies the starting timestamp. These lines are grouped based on their b_i and a_i values. To ensure successful decoding, binary counters are introduced before each group, denoted as $\#b$ and $\#a$. For each merged group, the list of starting timestaps of its constituent line segments is stored. Starting values b_i are quantized based on the maximum error threshold ($error$) used and the resulting integers are delta-encoded. This error, the first y-intercept (b_1) and the last timestamp of the timeseries (t_{last}) are encoded too. Sim-Piece+ is an enhanced version of the original Sim-Piece which further improves the compression ratio by taking advance of a Variable-Byte encoding technique [15]. Variable-Byte encoding is a simple and effective method used for integer compression. The encoding scheme works by representing an integer value using a variable number of bytes. The basic idea is to use the most significant bit of each byte to indicate whether it is the last byte in the representation or if there are more bytes to follow. In Sim-Piece+, the utilization of Delta encoding is incorporated alongside Variable-Byte encoding to compress starting timestamps. By combining these two techniques, the compression ratio is significantly improved.

3 EXPERIMENTAL EVALUATION

Figure 2 shows the average compression ratio achieved by Sim-Piece and state-of-the-art PLA techniques, i.e., PMC-MR [10], Swing and Slide [4] and Mixed [12], on a real dataset extracted from [13]. The maximum error threshold used was 0.5% and 5% of each dataset range of values. Sim-piece clearly outperforms all prior PLA techniques. Sim-piece+ manages to achieve even higher compression ratios, due to the enhanced data encoding it utilizes.

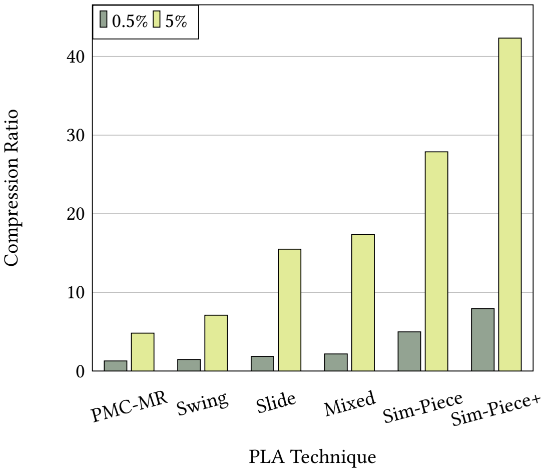

Figure 2: Average compression ratio comparison of lossy PLA approaches for two error thresholds.

REFERENCES

[1] Antonios Deligiannakis and Yannis Kotidis. 2005. Data reduction techniques in sensor networks. *IEEE Data Eng. Bull.*, 28, 19–25.

[2] Antonios Deligiannakis, Yannis Kotidis, and Nick Roussopoulos. 2007. Dissemination of compressed historical information in sensor networks. *The VLDB Journal*, 16, 4, 439–461. DOI: 10.1007/s00778-005-0173-5.

[3] Antonios Deligiannakis, Vassilis Stoumpos, Yannis Kotidis, Vasilis Vassalos, and Alex Delis. 2008. Outlier-aware data aggregation in sensor networks. In *Proceedings of ICDE 2008*, 1448–1450. DOI: 10.1109/ICDE.2008.4497585.

[4] Hazem Elmeleegy, Ahmed K. Elmagarmid, Emmanuel Cecchet, Walid G. Aref, and Willy Zwaenepoel. 2009. Online piece-wise linear approximation of numerical streams with precision guarantees. *Proc. VLDB Endow.*, 2, 1, 145–156. DOI: 10.14778/1687627.1687645.

[5] Nikos Giatrakos, Yannis Kotidis, Antonios Deligiannakis, Vasilis Vassalos, and Yannis Theodoridis. 2013. In-network approximate computation of outliers with quality guarantees. *Inf. Syst.*, 38, 8, 1285–1308. DOI: 10.1016/j.is.2011.08.005.

[6] Nikos Giatrakos, Yannis Kotidis, Antonios Deligiannakis, Vasilis Vassalos, and Yannis Theodoridis. 2010. TACO: tunable approximate computation of outliers in wireless sensor networks. In *Proceedings of ACM SIGMOD*, 279–290. DOI: 10.1145/1807167.1807199.

[7] S.L Hakimi and E.F Schmeichel. 1991. Fitting polygonal functions to a set of points in the plane. *CVGIP: Graphical Models and Image Processing*, 53, 2, 132–136. DOI: https://doi.org/10.1016/1049-9652(91)90056-P.

[8] Evgeny Kharlamov et al. 2019. An ontology-mediated analytics-aware approach to support monitoring and diagnostics of static and streaming data. *J. Web Semant.*, 56, 30–55. DOI: 10.1016/j.websem.2019.01.001.

[9] Xenophon Kitsios, Panagiotis Liakos, Katia Papakonstantinopoulou, and Yannis Kotidis. 2023. Sim-piece: highly accurate piecewise linear approximation through similar segment merging. *Proc. VLDB Endow.*, 16, 8, 1910–1922. DOI: 10.14778/3594512.3594521.

[10] Iosif Lazaridis and Sharad Mehrotra. 2003. Capturing sensor-generated time series with quality guarantees. In *Proc. of the 19th Int. Conf. on Data Engineering*, 429–440. DOI: 10.1109/ICDE.2003.1260811.

[11] Panagiotis Liakos, Katia Papakonstantinopoulou, and Yannis Kotidis. 2022. Chimp: efficient lossless floating point compression for time series databases. *Proc. VLDB Endow.*, 15, 11, 3058–3070. DOI: 10.14778/3551793.3551852.

[12] Ge Luo, Ke Yi, Siu-Wing Cheng, Zhenguo Li, Wei Fan, Cheng He, and Yadong Mu. 2015. Piecewise linear approximation of streaming time series data with max-error guarantees. In *2015 IEEE 31st International Conference on Data Engineering*, 173–184. DOI: 10.1109/ICDE.2015.7113282.

[13] National Ecological Observatory Network (NEON). 2022. en. (2022). https://data.neonscience.org/.

[14] Tuomas Pelkonen, Scott Franklin, Justin Teller, Paul Cavallaro, Qi Huang, Justin Meza, and Kaushik Veeraraghavan. 2015. Gorilla: a fast, scalable, in-memory time series database. *Proc. VLDB Endow.*, 8, 12, 1816–1827. DOI: 10.14778/2824032.2824078.

[15] Hugh E. Williams and Justin Zobel. 1999. Compressing Integers for Fast File Access. *The Computer Journal*, 42, 3, 193–201. DOI: 10.1093/comjnl/42.3.193.