Similarity-based Compression with Multidimensional Pattern Matching

Olivia Del Guercio
Lawrence Berkeley National Laboratory
Berkeley, CA
delgur@gmail.com

Rafael Orozco
Bucknell University
Lewisburg, PA
rao010@bucknell.edu

Alex Sim
Lawrence Berkeley National Laboratory
Berkeley, CA
asim@lbl.gov

Kesheng Wu
Lawrence Berkeley National Laboratory
Berkeley, CA
kwu@lbl.gov

ABSTRACT

Sensors typically record their measurements using more precision than the accuracy of the sensing techniques. Thus, experimental and observational data often contain noise that appears random and cannot be easily compressed. This noise increases storage requirement as well as computation time for analyses. In this work, we describe a line of research to develop data reduction techniques that preserve the key features while reducing the storage requirement. Our core observation is that the noise in such cases could be characterized by a small number of patterns based on statistical similarity. In earlier tests, this approach was shown to reduce the storage requirement by over 100-fold for one-dimensional sequences. In this work, we explore a set of different similarity measures for multidimensional sequences. During our tests with standard quality measures such as Peak Signal to Noise Ratio (PSNR), we observe that the new compression methods reduce the storage requirements over 100-fold while maintaining relatively low errors in PSNR. Thus, we believe that this is an effective strategy to construct data reduction techniques.

CCS CONCEPTS

• **Computing methodologies** → *Probabilistic reasoning*; *Feature selection*; • **Networks** → **Network performance analysis**.

KEYWORDS

Compression; Performance pattern; test statistics; `IDEALEM`

ACM Reference Format:
Olivia Del Guercio, Rafael Orozco, Alex Sim, and Kesheng Wu. 2019. Similarity-based Compression with Multidimensional Pattern Matching. In *Systems and Network Telemetry and Analytics (SNTA'19), June 25, 2019, Phoenix, AZ, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3322798.3329252

SNTA'19, June 25, 2019, Phoenix, AZ, USA

ACM ISBN 978-1-4503-6761-5/19/06…$15.00
https://doi.org/10.1145/3322798.3329252

1 INTRODUCTION

In this new age of data, the amount of available data increases faster than our capacity to analyze it. One of the highest priorities in data science is the ability to analyze streaming data from the large number of sensors; however, as the amount of data increases, real-time analysis becomes increasingly challenging. One of the solutions to this issue is to build algorithms that reduce incoming data into more manageable sizes. One common method for this is compression [16, 22, 24]. However, even the best-known lossy compression techniques do not work well on noisy sensor data [2, 12]. Typically, analysis tasks on such datasets focus on large-scale features or extreme values, and not on small variations. In these cases, we have developed an effective approach based on statistical similarity [6, 9–11].

On large datasets with mostly floating-point values, lossless compression techniques are not effective; thus, none of the recently developed compression methods for numerical values attempts to preserve the full precision of the original values. Those new techniques are categorized as lossy methods [1, 2, 7, 8, 12], and mmong them, ZFP [12] and SZ [2, 21] are particularly effective in taking advantage of the relatively slow variations among the neighbors in space and time. On large simulation datasets, where the phenomenon being simulated is captured in enough precision that the neighboring cells typically have adjoining values, both ZFP and SZ could reduce the storage requirements by a factor of over 100. However, in sensor data streams, such smoothness is not present. For example, the electric current from a power grid monitoring measurement dataset and the electric voltage in an electroencephalogram (EEG) both appear to be quite random. In such cases, these state of art floating-point compression algorithms are still not effective.

In these and many other use cases, small random fluctuations are not of interest to the domain scientists. Therefore, it is sufficient to capture the main statistical properties. In designing a new compression method, the key choice is what statistical properties should be preserved. Our initial work choose to preserve a statistical property known as exchangeability [4]. In a number of earlier tests, we showed that this technique was able to reduce the storage requirements by more than 100-fold while preserving important properties

of the data [9–11]. However, the existing software relies on a statistical similarity measure known as the Kolmogorov-Smirnov (KS) test that is only able to work with one-dimensional sequences.

The main contribution of this work includes:

- we explore new similarity measures that can be more easily extended to higher dimensions;
- we conduct extensive tests with a number of different types of data to study the effectiveness of the compression technique with the new similarity measures;
- we compare against the state of art methods using the common compression quality measure including Peak Signal to Noise Ratio (PSNR).

This work further demonstrates that the data reduction technique with statistical similarity is an effective approach for compression.

The rest of this paper is organized as follows. In Section 2, we briefly review related work and discuss the key design considerations of the software implementation known as IDEALEM [19]. We describe the new multidimensional similarity measures in Section 3. An extensive evaluation of IDEALEM is given in Section 4. We conclude with a brief summary and the discussion of future work in Section 5.

2 RELATED WORK

Data compression reduces the storage for representing the same information. This is accomplished by identifying patterns in the data [16]. Data compression methods are categorized into two broad classes: *lossless coding* where the reconstruction of compressed data is identical to the original data; and *lossy coding* where the reconstructed data is different from the original data. Next, we briefly review lossy compression methods and highlight two design considerations that drive the design of Implementation of Dynamic Extensible Adaptive Locally Exchangeable Measures (IDEALEM) [19]. The first one is redefining the distance (similarity) measure between sequences of numerical values; and the second is to allow analysis to be performed directly on the compressed data.

Redefine Similarity Measures. We focus on lossy techniques as they can more effectively reduce the storage requirement. For floating-point values, a common coding method is quantization [16]. In fact, the most effective compression techniques, such as ZFP [12] and SQE [7], are based on quantization. Another common approach is to apply some forms of prediction based on neighbors in space and time [2].

The information loss due to compression is often measured by the Euclidean distance (ℓ_2 distance) between reconstructed data and the original data. This distance may be represented as the mean squared error (MSE) and the signal-to-noise ratio (SNR) [14, 16]. To increase the possibility of compression, we believe it is necessary to adopt alternative statistical measures for differences between original data and compressed data. Earlier, we used the KS similarity measure; and in this work, we explore additional similarity measures. We note that none of these measures are strictly distance measures. The rationale behind this choice is that this choice increases the potential of data reduction by allowing more forms of differences to be tolerated. This is driven by the observation that the relatively small variations in the sensor data are not important to the scientific applications. For example, the exact voltage measurements are not of interest to electric grid system operators as long as they are within the specified tolerance.

Though the KS similarity measure was found to be effective for one-dimensional sequences [9–11], we have not found an easy way to extend it to multidimensional data. In this work, we explore similarity measures that could be more easily extended to multidimensional data [18], as shown in Section 3.

Allowing Analysis without Decompression. Our compression method IDEALEM stores the first instance of each group of similar data block as is. These preserved data blocks act as dictionary entries of a dictionary-based compression method [16, 20]. Similar to other dictionary-based methods, the compressed data can be used directly without decompression, which allows advanced analysis operations directly on the compressed data. In this work, we focus on the basic properties of the decompressed data, such as PSNR.

By storing the first instance of each group of similar blocks precisely, we can reproduce this block exactly when we encounter it for the first time during decompression. If this data block only appears once, it is preserved accurately. Typically, the data blocks containing extreme values are distinct from others and therefore would be preserved with our compression method.

When a dictionary block is used for a second time, we need to decide what to do with the actual values. We have studied a number of options to mitigate the impact of reproducing multiple copies from a single dictionary block [23]. In general, we can regard this as regenerating data from a recorded kernel or generator. This suggests a number of different analysis options that we plan to explore in the near future.

3 MULTIDIMENTIONAL SIMILARITY MEASURES

Given two sequences of n values each, $\mathbf{x} \equiv (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} \equiv (y_1, y_2, \ldots, y_n)$, the most common way to measure the difference between them is the Euclidean distance

$$E = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \tag{1}$$

There are a number of different ways to generalize this difference measure (also known as dissimilarity) [18]. We broadly refer to them as similarity measures. From earlier studies [17, 18], we have selected the dynamic time warping (DTW) [15] and minimum jump cost (MJC) [17] as the similarity measures to study for this work. Previous publications have shown that these similarity measures are effective in a wide variety of applications and are also relatively easy to compute [17, 18]. For clarity, we only describe these similarity measures for one-dimensional sequences, however, these similarity measures can be extended easily to multidimensional cases where each of x_i and y_i is a vector.

Dynamic Time Warping. Dynamic Time Warping (DTW), see Fig. 1, is a classic definition of similarity between $\mathbf{x}$ and $\mathbf{y}$. Let $D_{j,k}$ denote the similarity between the first j elements of $\mathbf{x}$ and the first k elements of $\mathbf{y}$. Starting with $D_{0,0} \equiv 0$, the DTW measure $D_{j,k}$ is recursively defined as

$$D_{j,k} = f(x_j, y_k) + \min\left\{D_{j,k-1}, D_{j-1,k}, D_{j-1,k-1}\right\}, \tag{2}$$

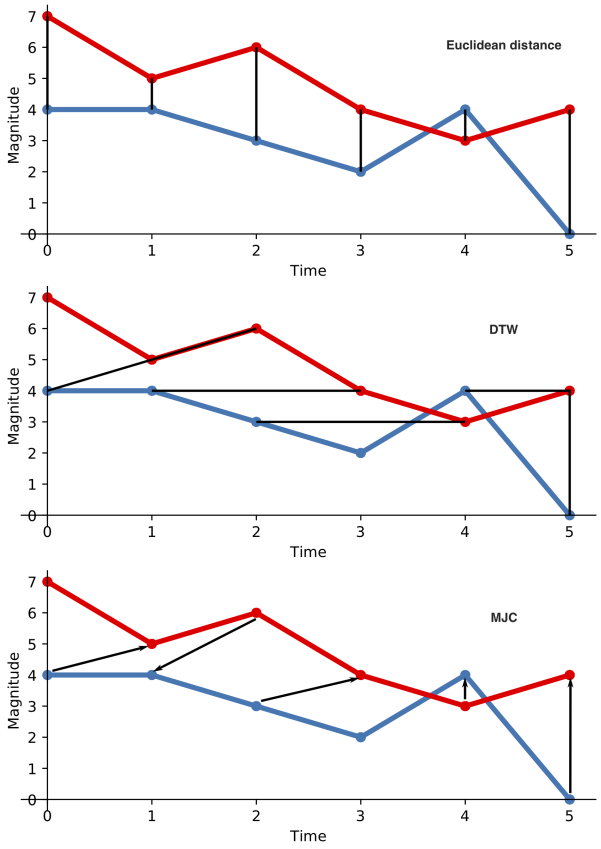

Figure 1: Distance measures: Euclidean distance, dynamic time warp (DTW) and minimum jump cost (MJC) with respective similarity measures of 14, 11, and 11.

The function $f(x_j, y_k)$ in Equation 2 is known as the sample similarity function, which is often taken to the Euclidean distance. We follow this practice in the current work.

It is necessary to compute all n^2 values of $D_{j,k}$ in order to obtain the final DTW measure $D_{n,n}$. This matrix of $D_{j,k}$ also provides a way for aligning **x** and **y**.

Minimum Jump Cost. Minimum jump cost (MJC) [17] works by accumulating the cost of "jumping" from one time series data point to the nearest data point in the other time series.

$$J = \sum_i C^{(i)}_{min} \equiv \sum_i \min\left\{c^{y_j}_{x_i}, c^{y_{j+1}}_{x_i}, \ldots\right\}, \tag{3}$$

where $j-1$ is the position in **y** that x_{i-1} jumped to, and $c^{y_j}_{x_i}$ is the sample similarity measure that includes the time difference and the difference between x_i and y_j [17]. An illustration of MJC is given in Fig. 1.

MJC preserves the core concept of DTW, but no longer computes all n^2 similarity values to obtain the final one. Therefore, we expect MJC to require less compute time, and what we need to understand is how well it works for defining similar data blocks for data reduction.

Both DTW and MJC give up the need to match the ith element of **x** with the ith element of **y**, which effectively allows values in two sequences to appear in different orders. This is an important feature in our attempt to relax the similarity measure for data reduction, and also motivated our choices on how to reconstruct the data blocks from dictionary blocks [23].

4 EVALUATION

In this section we present an empirical evaluation of IDEALEM with the two similarity measures, and explore how well they work for multidimensional data compression. We start with a brief description of the test datasets.

4.1 Multidimensional Datasets

Power grid monitoring measurements. This dataset is from an ongoing experiment at Lawrence Berkeley National Laboratory (LBNL). It contains data from μPMU installed around LBNL site. The full dataset contains twelve variables related to four measurements about the three phases of the alternate current system [13]: voltage, current, phase angles of the voltage, and phase angle of the current. Some individual measurements have been used in previous studies [9, 11].

Distributed Acoustic Sensing. Distributed Acoustic Sensing (DAS) is a technology that turns unused optical fibre meant for telecommunication into sensors for ground motion [5]. DAS data can be viewed as generated from thousands of sensors, which we regard as thousands of dimensions for our time series data. In this test, we have used a sample DAS dataset containing 11,000 sensors over 30000 time points [5], which is from a larger dataset collected over a three-week monitoring period (April 4-26, 2015) resulting about 2.7 terabyte in size and 31,000 individual files.

Natural images. In this study we consider the compression of full color images. We use images in public domain that contain a large range of colors. For example, we use the scene of a sunset in Fig. 4 which displays vivid reds that contrast with vivid blues. These kind of images should be in principle harder to compress since the underlying data is more varied. We format the image data as a 3×N vector where N is the number of pixels in the image. With this structure, each element of the vector is a 3d data point containing the red, green and blue integer values of a certain pixel.

4.2 Performance measures

To measure the effectiveness of IDEALEM, we focus on two common performance metrics: compression ratio (CR) and peak signal-to-noise ratio (PSNR), where CR is defined to be the ratio between uncompressed data size and compressed size, and PSNR can be defined as

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_x^2}{MSE}\right).$$

Where MSE is the mean squared error, and is defined as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2.$$

PSNR is expressed in terms of a logarithmic decibel scale, and is thus normalized even in datasets with large ranges. In the previous studies, we have avoided using PSNR and other quality measures based on MSE since IDEALEM is not designed to control Euclidean distance based data quality. However, in the realm of data compression, PSNR is one of the most commonly used quality measures, so we choose the PSNR vs. CR curve to display the compression effectiveness. It should be noted that PSNR relies on inverse MSE, so as PSNR increases, compression quality also increases. Values higher than 30 are generally considered to produce reliable decompression results.

4.3 MJC vs DTW

To choose between MJC and DTW, we use a subset of the power grid monitoring measurement dataset. We use the first two current magnitudes making it a 2D dataset. We select the parameters that allow the two compression algorithms to produce a compression ratio of 100, and compare the PSNR and execution time. The result is in Table 1. We expected the compression algorithm with MJC as similarity measure to use less CPU time, while sacrificing quality to a certain extent. We observed that for a 8× speed-up, there would only be a 2% decrease in PSNR.

We expected MJC to have a faster run-time than DTW because it only jumps forward and doesn't have to compute the full matrix multiplication. We also observed that MJC has lower error for the full dictionary size as shown in Table 1.

Dictionary Size		PSNR	Run-time	MSE
2	DTW	32.4	0.214	0.281
	MJC	32.6	0.183	0.262
20	DTW	34.4	0.624	0.215
	MJC	34.3	0.336	0.217
100	DTW	35.5	1.339	0.0580
	MJC	35.4	0.629	0.0586
255	DTW	34.8	1.318	0.0580
	MJC	35.4	0.686	0.0586

Table 1: A comparison of PSNR, run-time, and MSE at 100 CR for various dictionary sizes of MJC and DTW.

We can summarize that in general, MJC outperforms DTW. When analyzing LBNL power grid monitoring dataset with the largest dictionary size, we also observed the effectiveness of MJC against the leading $O(n)$ time compression algorithm in SZ [2, 3, 21]. In general, SZ [2, 21] is efficient and results in better compression ratios. However, for those highly variable data, SZ leads to lower PSNR due to its dependency on nearby points to perform reconstruction, such as output from power grid monitoring sensors as shown Fig. 3. SZ minimizes the MSE of the data which leads to data in high variable ratios to be over-simplified to show only the mean of nearby data points. IDEALEM has an intrinsic difference with statistical similarity measures, where the MSE is controlled indirectly with a similarity measure and by using stored buffers of actual data. This results in the decompressed data closer to the original dataset especially in highly variable dataset. In Fig. 3, MJC visually deviates much less from the original data than SZ at 100 CR.

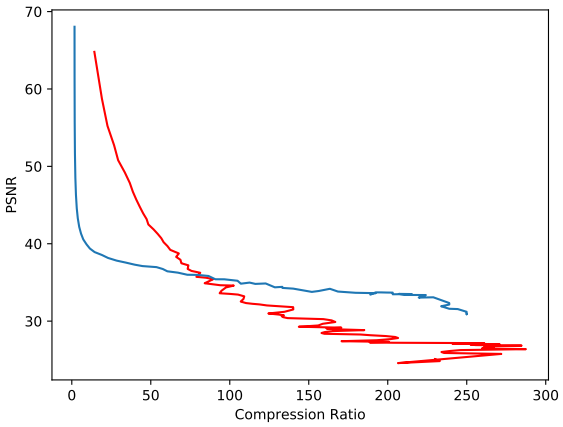

Figure 2: Compression ratio plotted against PSNR for SZ (red) and MJC (blue)

Also, in Fig. 2, MJC has higher PSNR (lower error) at compression ratios above 80. While there is a relative inefficiency in run-time, MJC preserves many features in reconstructed (decompressed) data especially for highly variable data.

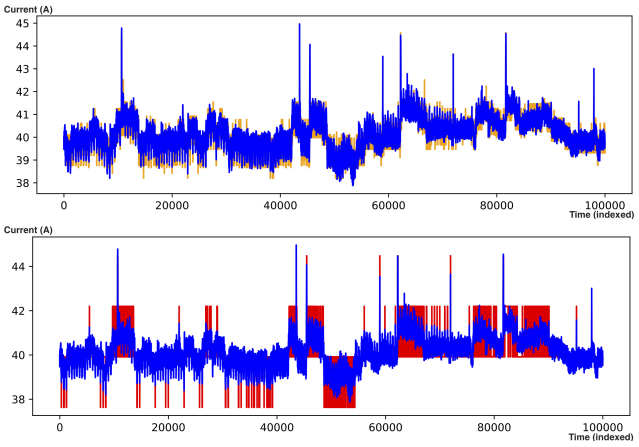

Figure 3: Comparison of MJC (top figure) and SZ (bottom figure) compression quality over a single stream of power grid monitoring data for current over time at 100 CR. MJC has a run-time of 0.686 seconds with a PSNR of 35.4, and SZ has a run-time of 0.009 seconds with a PSNR of 33.5.

4.4 Visual Validation

As a preliminary validation of our compression approach, we show some comparison between compressed and uncompressed images. In this case, we treat RGB values of pixels as dimensions and linearize the pixels in a time series, and an example on a sample image

Figure 4: Compressing an image as 3D (RGB) sequences could reduce the storage requirement significantly with minimal compression artifacts.

[1] is shown in Fig. 4. We can visually see that the compressed images in Fig. 4 contains minimum compression artifact even at the compression ratio above 50.

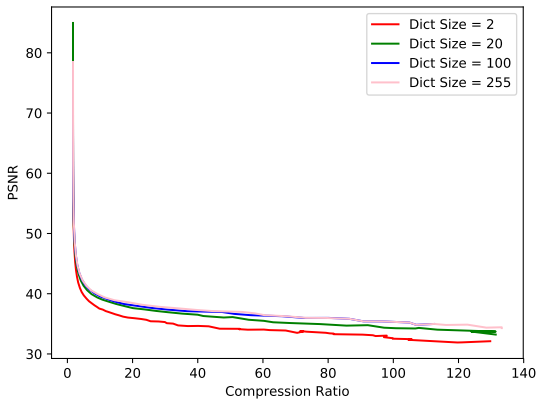

Figure 5: Impact of the dictionary size. Generally, a larger dictionary size is better for compression, but in this case, increasing dictionary size does not significantly change either CR or PSNR.

4.5 Selecting IDEALEM Parameters

IDEALEM compression is controlled by a number of parameters [6], and we briefly illustrate next how they affect CR and PSNR. We perform the following tests using the two dimensional power grid monitoring dataset. Fig. 5 shows how the size of the dictionary affects the performance of IDEALEM. Generally, we see that with a modest dictionary size, the PSRN vs. CR curves are about the same, which indicates that the number of distinct patterns are relatively small. Had there been more distinct patterns, more dictionary entries might be needed in order to achieve better compression ratio or larger PSNR.

[1]image by Nina Fox, https://chemtrailsnorthnz.wordpress.com/2010/07/23/hawaii-napili-sunset%E2%80%8F-july-21-2010/

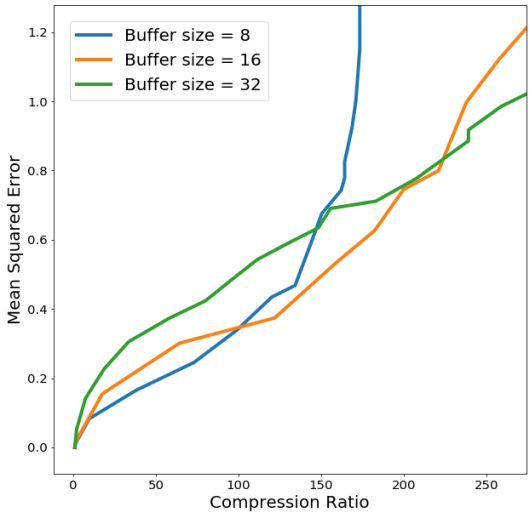

Figure 6: Impact of the buffer size. Generally smaller buffer sizes produced higher PSNR.

Fig. 6 illustrates how the buffer size affects the performance. While there is no theoretical understanding of how to select this parameter, we observe from Fig. 6 that the buffer size of 8 produces better data quality in compression ratios of 0-100, and a buffer size of 16 produces better data quality at compression ratios 100-225. At larger compression ratios, 32 buffer size performs the best. From this initial analysis, we can give the preliminary conclusion that at smaller compression ratios, smaller buffer sizes work better and that at larger compression ratios, it is better to use larger buffer sizes.

5 SUMMARY AND FUTURE WORK

In this paper, we describe the multidimensional similarity measures as well as their impact on the data reduction. This work extends the IDEALEM, an approach of using statistical pattern matching to reduce the storage requirement for sequences of numerical values to utilizes multidimensional similarity measures including Minimum Jump Cost (MJC) and Dynamic Time Warp (DTW) to find similar data blocks and create a dictionary of "patterns." By using these similarity measures, we are able to apply the statistical similarity based data reduction technique to multidimensional sequences. For our tests, we evaluated with a variety of data including sensor data, images and video (not described in the paper) that it is possible to reduce the storage requirement by more than 100-fold while preserving essential features of the data. The work is incorporated in a software package called IDEALEM and is available at [19].

There are considerable amount of additional work need to further quantify the effectiveness of this approach. For example, there are

many different ways of selecting the similarity measures. We have demonstrated that some of them work quite well, and we need to develop a framework to automatically select the most effective similarity measure. For high dimensional data (e.g. video), there are many ways of folding the data into sequences, and additional work is needed to understand the optimal data organization in these cases.

ACKNOWLEDGMENT

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This research used resources of the National Energy Research Scientific Computing Center.

REFERENCES

[1] Martin Burtscher and Paruj Ratanaworabhan. 2009. FPC: a high-speed compressor for double-precision floating-point data. *IEEE Trans. Comput.* 58, 1 (Jan. 2009), 18–31.

[2] Sheng Di and Franck Cappello. 2016. Fast error-bounded lossy HPC data compression with SZ. In *Proc. Int'l Parallel Distrib. Process. (IPDPS '16)*. 730–739.

[3] Sheng Di, Dingwen Tao, and Franck Cappello. 2017. SZ: fast error-bounded floating-point data compressor for scientific applications. Retrieved February 3, 2017 from https://collab.mcs.anl.gov/display/ESR/SZ

[4] Persi Diaconis. 1988. Recent progress on de FinettiâĂŹs notions of exchangeability. *Bayesian statistics* 3 (1988), 111–125.

[5] Shan Dou, Nate Lindsey, Anna M. Wagner, Thomas M. Daley, Barry Freifeld, Michelle Robertson, John Peterson, Craig Ulrich, Eileen R. Martin, and Jonathan B. Ajo-Franklin. 2017. Distributed Acoustic Sensing for Seismic Monitoring of The Near Surface: A Traffic-Noise Interferometry Case Study. *Nature* (September 2017), 11620. https://doi.org/10.1038/s41598-017-11986-4

[6] Kade Gibson, Dongeun Lee, Jaesik Choi, and Alex Sim. 2018. Dynamic Online Performance Optimization in Streaming Data Compression. In *Proc. IEEE International Conference on Big Data (Big Data 2018)*.

[7] Jeremy Iverson, Chandrika Kamath, and George Karypis. 2012. Fast and effective lossy compression algorithms for scientific datasets. In *Proc. Int'l Conf. Parallel Process. (Euro-Par '12)*. 843–856.

[8] Sriram Lakshminarasimhan, Neil Shah, Stephane Ethier, Scott Klasky, Rob Latham, Rob Ross, and Nagiza F. Samatova. 2011. Compressing the incompressible with ISABELA: in-situ reduction of spatio-temporal data. In *Proc. Int'l Conf. Parallel Process. (Euro-Par '11)*. 366–379.

[9] Dongeun Lee, Alex Sim, Jaesik Choi, and Kesheng Wu. 2016. Novel data reduction based on statistical similarity. In *Proc. Int'l Conf. Scient. Stat. Database Manag. (SSDBM '16)*. 21:1–21:12.

[10] DongEun Lee, Alex Sim, Jaesik Choi, and Kesheng Wu. 2017. Expanding Statistical Similarity Based Data Reduction to Capture Diverse Patterns. In *2017 Data Compression Conference (DCC)*. 445–445. https://doi.org/10.1109/DCC.2017.77

[11] Dongeun Lee, Alex Sim, Jaesik Choi, and Kesheng Wu. 2017. Improving Statistical Similarity Based Data Reduction for Non-Stationary Data. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management (SSDBM '17)*. ACM, New York, NY, USA, Article 37, 6 pages. https://doi.org/10.1145/3085504.3085583

[12] Peter Lindstrom. 2014. Fixed-Rate Compressed Floating-Point Arrays. *IEEE Trans. Vis. Comput. Graphics* 20, 12 (Dec. 2014), 2674–2683.

[13] Sean Peisert, Reinhard Gentz, Joshua Boverhof, Chuck McParland, Sophie Engle, Abdelrahman Elbashandy, and Dan Gunter. 2017. *LBNL Open Power Data*. Technical Report. Lawrence Berkeley National Laboratory.

[14] Iain E Richardson. 2010. *The H.264 Advanced Video Compression Standard* (second ed.). John Wiley and Sons.

[15] Hiroaki Sakoe and Seibi Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing* 26, 1 (1978), 43–49.

[16] Khalid Sayood. 2012. *Introduction to data compression* (fourth ed.). Newnes.

[17] Joan Serrà and Josep Lluís Arcos. 2012. A Competitive Measure to Assess the Similarity between Two Time Series. In *Case-Based Reasoning Research and Development*, Belén Díaz Agudo and Ian Watson (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 414–427.

[18] Joan SerrÃă and Josep Ll. Arcos. 2014. An empirical evaluation of similarity measures for time series classification. *Knowledge-Based Systems* 67 (2014), 305–314. https://doi.org/10.1016/j.knosys.2014.04.035

[19] Alex Sim, Dongeun Lee, Kesheng Wu, and Jaesik Choi. 2016. IDEALEM. https://sdm.lbl.gov/idealem.

[20] Przemysław Skibiński, Szymon Grabowski, and Sebastian Deorowicz. 2005. Revisiting dictionary-based compression. *Software: Practice and Experience* 35, 15 (2005), 1455–1476.

[21] D. Tao, S. Di, Z. Chen, and F. Cappello. 2017. Significantly Improving Lossy Compression for Scientific Data Sets Based on Multidimensional Prediction and Error-Controlled Quantization. In *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. 1129–1139. https://doi.org/10.1109/IPDPS.2017.115

[22] J. Uthayakumar, T. Vengattaraman, and P. Dhavachelvan. 2018. A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications. *Journal of King Saud University - Computer and Information Sciences* (2018). https://doi.org/10.1016/j.jksuci.2018.05.006

[23] Kesheng Wu, Dongeun Lee, Alex Sim, and Jaesik Choi. 2017. Statistical data reduction for streaming data. In *2017 New York Scientific Data Summit (NYSDS)*. 1–6. https://doi.org/10.1109/NYSDS.2017.8085035

[24] Jacob Ziv and Abraham Lempel. 1977. A universal algorithm for sequential data compression. *IEEE Trans. Inf. Theory* 23, 3 (May 1977), 337–343.