Statistical Data Reduction for Streaming Data

Kesheng Wu, Dongeun Lee, Alex Sim, and Jaesik Choi

***Abstract*—Bulk of the streaming data from scientific simulations and experiments consists of numerical values, and these values often change in unpredictable ways over a short time horizon. Such data values are known to be hard to compress, however, much of the random fluctuation is not essential to the scientific application and could therefore be removed without adverse impact. We have developed a compression technique based on statistical similarity that could reduce the storage requirement by over 100-fold while preserve prominent features in the data stream. We achieve these impressive compression ratios because most data blocks have similar probability distribution and could be reproduced from a small block. The core concept behind this work is the exchangeability in statistics. To create a practical compression algorithm, we choose to work with fixed size blocks and use Kolmogorov-Smirnov test to measure similarity. The resulting technique could be regarded as a dictionary-based compression scheme. In this paper, we describe the method and explore its effectiveness on two sets of application data. We pay particular attention to the Fourier components of the reconstructed data and show that in addition to preserving unique features in data it is also faithfully preserving the Fourier components whose periods extend more than a few blocks.**

I. Introduction

In science, computerized simulation and monitoring systems are producing petabytes of data right now [1], [2], and their data production rates are increasing. This creates significant challenges for data management and data analysis. One common tool for addressing the data volume issue is compression [3], [4]. The bulk of such data is floating-point values, which are known to be particularly hard to compress, even for lossy compression techniques [5], [6] because these values contain small but unpredictable variations in space and time. Often the analysis tasks on the data focus on the large-scale features of the data, not the small variations. In such cases, capturing the large-scale statistical properties correctly would be sufficient. In this work, we aim to devise one such compression technique that can preserve the large-scale features in the data while only preserving some statistical properties at the fine-scale.

On large datasets with mostly floating-point values, the lossless compression techniques typically can not reduce the storage requirement significantly, therefore, none of the recent developed compression methods for numerical values attempts to preserve the full precision of the original values. These techniques are lossy [7], [5], [8], [9], [6]. Among them, ZFP [6] and SZ [5] are particularly effective in taking advantage of the relatively slow variations of the neighboring values in space and time. They can both reduce the storage requirements by a factor of over 100 on large simulation datasets becaue the phenomenon being simulated are captured in enough precision that the neighboring cells typically have adjoining values. However, in many sensor data streams, such smoothness is not present, for example, the electric current from a power grid monitor 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 the above mentioned example datasets, we note that the small random fluctuations are not of interest to the domain scientists. Therefore, it is sufficient to capture some key statistical properties of the data. In designing a new compression method, the key choice what statistical properties should be preserved. The basic statistical concept we plan to follow is called exchangeability [10]. To make this general theoretical construct usable in a computer algorithm, we propose to break the incoming data stream into blocks and adopt a simple statistical test to measure the similarity between two blocks. We call our similarity measure the Locally Exchangeable Measure or LEM for short [11]. As we show later, there is a very effective way to implement our LEM based compression technique.

The statistical test used in this work is Kolmogorov-Smirnov (KS) test, which effectively compares the cumulative distribution of two input sequences. Thus, the similarity preserved in this new technique is the empirical probability distribution of the data block. When two blocks' probability distributions are about the same (according to KS test), they are declared to be the same, and one of them would be reproduced from the other.

Traditionally, the difference between the compressed data and the original data is measured using their Euclidean distances, and a method that produces smaller distances is better than the one that produces larger distances. Our new compression method does not attempt to keep a small distance between the compressed data sequence and original data sequence, but instead attempts to keep a small distance between the probability distributions of the two sequences. This is a significant departure from the common practice in designing compression techniques.

We have described an initial study of this technique recently [12]. That study was limited to examine the compression ratio with a set of electric power grid data. Though we will briefly examine the same data set in this work. One key objective of this work is to demonstrate our approach can preserve important properties such as the extreme values and the Fourier spectrum of the input data. We also use a new data set from neural science to demonstrate that the effectiveness of our approach is not limited to one application.

The rest of this paper is organized as follows. In Section II, we briefly review related work and discuss the key design considerations of the new algorithm. In Section III, we provide a brief overview of the IDEALEM implementation and give some suggestions on how to choose the key parameters. An extensive evaluation of IDEALEM is given in Section IV. We conclude with a brief summary and the discussion of future work in Section V.

II. New Strategy for Data Compression

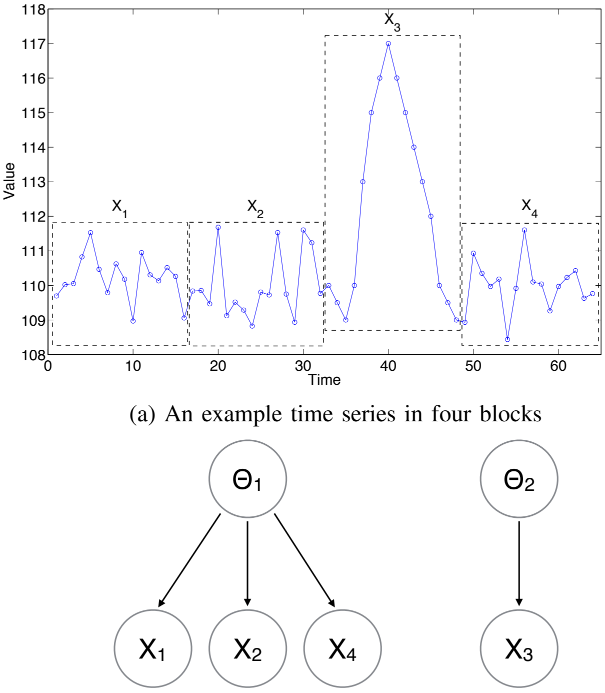
Data compression reduces the storage required for representing the same information. This is accomplished by identifying patterns in the data [3]. Data compression methods are categorized into two broad classes: *lossless coding* where the reconstruction of compressed data is identical to the original data; and *lossy coding* where the reconstructed data is different from the original data. Next, we briefly review related compression methods and highlight two design considerations that drive our work on the new compression method named IDEALEM (Implementation of Dynamic Extensible Adaptive Locally Exchangeable Measures) [13]. The first one is redefining the distance (similarity) measure to relax the order of values and to increase the possibility of compression; and the second is to allow analysis to be performed directly on the compressed data.

a) Relaxing Order of Values: Since the lossy compression techniques are better at reducing the storage requirement, we focus on lossy techniques. For floating-point values, a common coding method is quantization [3]. Many of the most effective compression techniques, such as ZFP [6] and SQE [8], are based on quantization. Another common approach is to apply some forms of prediction based on neighbors in space and time [5].

The information loss due to compression is generally 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) or the signal-to-noise ratio (SNR) [14], [3]. One fundamental limitation of this approach is that the order of the values is preserved. In some applications, the order of these values is not important, such as the electric current on a power grid. Giving up on preserving the order of the incoming values should lead to better compression.

b) Allowing Analysis without Decompression: Our LEM based 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 compression method [15], [3]. Similar to all other dictionary-based compression methods, the compressed data can be used directly without decompression. This is a useful property we plan to exploit in the future for more advanced analysis operations. In this work, we focus on the basic properties of the compressed data, such as preserving extreme values.

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. By this design choice, if this block only appears once, it is preserved accurately. Typically, the blocks containing extreme values are distinct from others and therefore would be preserved with IDEALEM compression.



(a) An example time series in four blocks

(b) Graphical model

Fig. 1: An illustration of IDEALEM compression method. Blocks X_1, X_2, and X_4 were generated from the normal distribution $\mathcal{N}(110, 1) \equiv \Theta_1$, while block X_3 is from a different generator Θ_2.

When a dictionary block is used a second time, we need to decide what to do with the actual values. One simple choice is to repeat the same values in the same order. However, as we will see later, this choice leaves artifacts in the Fourier spectrum. To avoid these artifacts, we perform a random shuffle of the values in dictionary block.

III. Outline of IDEALEM

Fig. 1a shows time series data of total 64 samples. If we assume that each sequence of 16 samples is an instantiation of a random variable X_i $(i = 1, \ldots, 4)$, we can consider similarities between these random variables. In Fig. 1a, X_1, X_2, and X_4 look similar; whereas X_3 looks different from other random variables. The design of IDEALEM is based on these observations: we may represent X_1, X_2, and X_4 using a single random variable, assuming that the three random variables have an identical distribution.

Fig. 1b displays the graphical model representation of the observations shown in Fig. 1a. We conceive a latent random variable Θ_j $(j = 1, 2)$ that governs random variables sharing the common distribution. In this paper, we focus on a practical data compression scheme leveraging the identical distribution shared by random variables with the same parent Θ_j, rather than consider relationships between these latent variables and infer the exchangeability of a new random variable for dynamic sampling, as discussed in the previous work [11], [12].

Specifically, if we keep only a single sequence (distribution) from X_1, X_2, and X_4, we can achieve compression ratio of 3, where the compression ratio is defined to be the ratio of the original size over the compressed size.

Given a sequence of values, IDEALEM breaks the sequence into fixed-size blocks and then test whether a new block is possibly produced from the same generator that produced one of the earlier blocks. In the current software implementation, this test is performed with Kolmogorov-Smirnov (KS) test. The output from a KS test is a score indicating the likelihood that the two sequences being compared are drawn from the same distribution, which we take to be the likelihood that the two data blocks are generated by the same generator. The user is expected to provide the threshold α for KS test, which can significantly affect the effectiveness of compression method overall. When the user neglects to provide one, the default value is 0.05, which is commonly used in many applications.

Other than block size and KS test threshold α, another important parameter users have to decide is the number of dictionary entries to be kept active during the compression and decompression process. The current implementation of IDEALEM uses a byte to address the entries in this dictionary, which limits the dictionary size to no more than 255 [13]. Let B denote the block size and D denote the dictionary size (number of dictionary entries). The total memory required by this dictionary is proportional to $B \times D$. To limit the size of this dictionary, we should keep the product of B and D relatively small. However, having a larger D typically means there is a higher likelihood of finding a match for a new block, which implies that we should a larger D. Since the limit imposed by our software is 255, which is small enough, we generally recommend users to keep D as 255. This is the value we use in the evaluations presented in the next section. In this set of tests, we keep the value of B as 32.

IV. Evaluation

Next we present an empirical evaluation of IDEALEM with the goal of demonstrating its usefulness in some cases. One of common criteria used to measure the effectiveness of a data reduction method is the compression ratio, defined to be the ratio between the original storage requirement and the compressed storage requirement. The second quality measure we use is the Fourier spectra of the compressed data. In addition, we will also provide evidence that IDEALEM preserves the extreme values in the original data, as discussed in Section II. We will not discuss conventional quality measures such as the mean squared error or the signal-to-noise ratio because IDEALEM is not designed to control the Euclidian distances between the original data and the compress data.

For this evaluation work, we use two sample datasets from two different application domains: electric power grid and neural science. The electric power grid monitoring dataset is from a device known as micro-Phaser Measurement Unit, or μPMU [16], [17], which records the voltage, current as well as their phase angles at milliseconds time intervals. These devices are usually installed at power transformers to monitor the health of the power grid system. The particular sample μPMU data has 5.3 million records and was collected at a transformer onsite at Lawrence Berkeley National Lab.

The second dataset is a sample of intracranial electroencephalogram (EEG) (also known as electrocorticography) [18]. We refer to it as the EEG dataset. This dataset consists of about 800,000 data values from a single channel of an EEG recording without patient information. We examine the Fourier spectrum of the EEG data because it is a property important to the neural science and engineering [18], [19].

A. *Compressing μPMU Data*

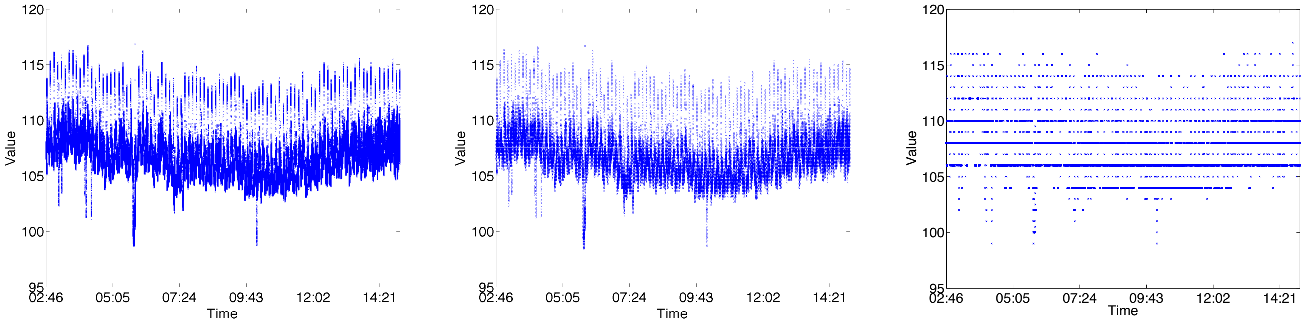
As a reference, we first provide the results of one of the best floating-point compression method ZFP [6], [20]. Similar to IDEALEM, ZFP is a lossy compression method. In numerous performance tests [6], ZFP was found to outperform all other compression methods, particularly for scientific simulation data in 3D arrays. On these 3D arrays, ZFP achieved compression ratios of 100 without noticeable information loss. These 3D datasets are from high-resolution simulations of physical phenomenon were the neighboring mesh points have similar values. Typically, on lower dimensional arrays, such as the 1D array used to represent the time series in the μPMU data, there are fewer neighbors to take advantages of and therefore less opportunity for compression. As shown in Fig. 2, the observed compression ratios are much less than 100.

Fig. 2a shows actual values of the μPMU dataset. Fig. 2b and Fig. 2c shows the results of ZFP compression with two different parameter setting.[1] It is clear that a large accuracy tolerance leads to more information loss and higher compression ratio. In the limit where a very large accuracy tolerance is used, eventually all reconstructed data values are set to zero and the corresponding compression ratio reaches 21.3.

Fig. 3 shows results of IDEALEM compression on the same data used in Fig. 2. The three plots shown three different KS test thresholds α. Clearly, this threshold significantly affects the compression ratios. With the default threshold value of 0.05, the compression ration is over 100.

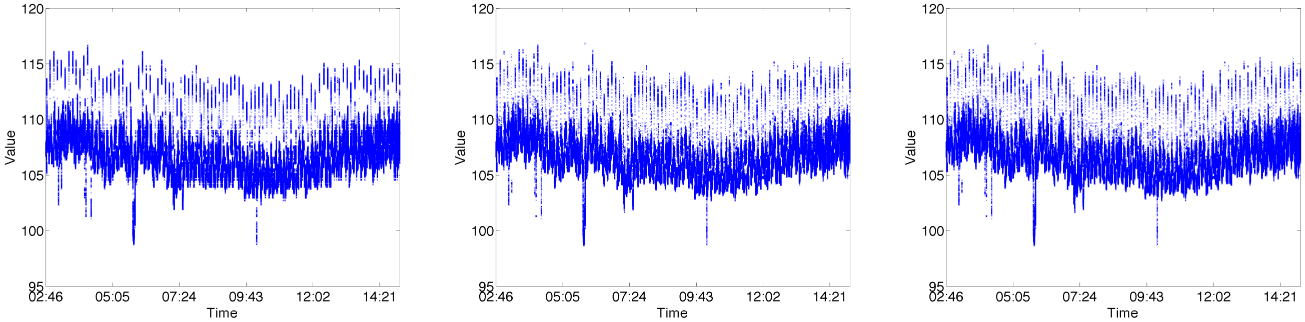
Another important observation from Fig. 3 is that there are no apparent compression artifacts, while there are clearly visible compression artifacts in the results of ZFP compression shown in Fig. 2b and 2c, even though ZFP achieves much lower compression ratios. The band structure shown in Fig. 2c is the distinctive signature of quantization artifact. Since IDEALEM always uses the values that actually appeared in the original data, it does not have any problem with quantization. Furthermore, its design also preserves extreme values as explained in Section II, which are often the most noticeable features in a dataset.

[1] The fixed-accuracy mode (option -a) was used [20], which specifies the maximum absolute difference between an uncompressed value and a reconstructed value.



(a) Sample μPMU data (b) ZFP `-a 1` (c) ZFP `-a 8`

Fig. 2: Scatter plots of one time series of the μPMU data. ZFP compression ratios are 7.5 in Fig.2b and 9.1 in Fig. 2c, both are much less than 100 achieved for 3-D simulation data [6].



(a) IDEALEM $\alpha = 0.01$ (b) IDEALEM $\alpha = 0.05$ (c) IDEALEM $\alpha = 0.1$

Fig. 3: Compression ratios of IDEALEM with with the above parameters are (a) 189.29, (b) 111.08, and (c) 86.91, respectively. They are considerably higher than with ZFP, and without the visible compression artifacts.

B. Compressing EEG Data

Fig. 4 shows the EEG dataset along with the compressed versions from ZFP and IDEALEM. As with the μPMU dataset, we again see that IDEALEM can achieve much higher compression ratio without creating visible compression artifact. More specifically, with an accuracy tolerance that is already producing visible quantization bands ZFP is only able to achieve a compression ratio of 12.6, while IDEALEM easily achieves a compression ratio of 106.6 without producing noticeable compression artifact. Again, IDEALEM compressed data requires less storage space and is able to represent the original data more faithfully.

With this set of EEG data, the application scientists want to preserve the Fourier spectrum for their analysis tasks [18], [19]. Therefore, we next examine the spectra of the original data and the compressed data. As shown in Fig. 5, the left halves of three spectral lines are the same, where a period of these Fourier components may span many data points, i.e., the large-scale features, are preserved well under IDEALEM compression. By construction, IDEALEM replaces values within a data block, and therefore would significantly alter Fourier components whose periods span a couple of blocks or a few points. For the case shown in Fig. 5, the block size is 32. We see that the Fourier components with frequencies less than 1/32 (~ 0.03) are the same in all three plots.

On the right side of Fig. 5, we see that the compressed data have different Fourier components than the original data. In particular, when we simply repeat the data in the dictionary blocks, the Fourier components of the compressed data clear contain artifacts due to the repeating values (bottom plot with label “NoShuffle” in Fig. 5); while the version of IDEALEM that shuffles the values in the dictionary blocks can avoid the obvious artifacts as shown in the middle plot labelled “Shuffle” in Fig. 5. Even though this reproduction of the high frequency components are not perfect, we note that these components are many orders of magnitude smaller than the lower frequency components, which indicates that they might be less important than the lower frequency components.

V. Conclusions

We propose a new way to construct data reduction techniques based on the statistical concept known as exchangeability. For each group of exchangeable data blocks, one representative data block is stored, which effectively creating a dictionary-based compression method. In this work, we report our experience of designing and implementing a concrete version of this dictionary-based compression method named IDEALEM. This method breaks an incoming data stream into fixed-size blocks and represents similar blocks with a one that

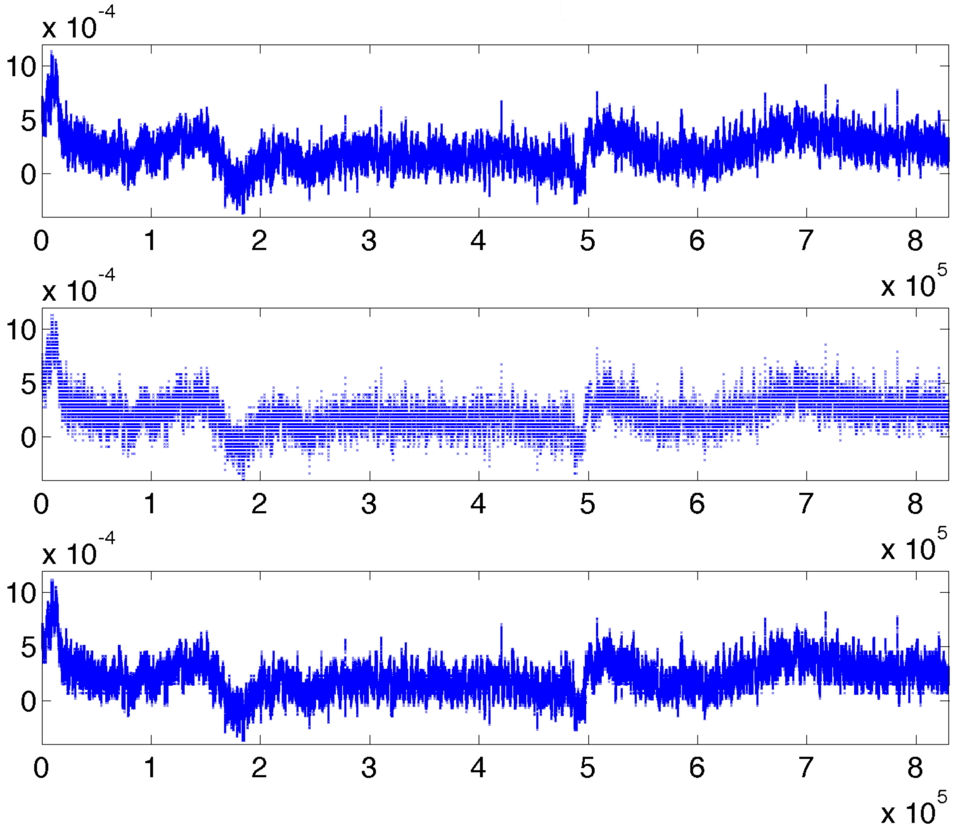


Fig. 4: Scatter plot of a sample EEG data, top: original data; middle: ZFP `-a 0.0004`, compression ratio 12.6; bottom: IDEALEM, compression ratio 106.6.

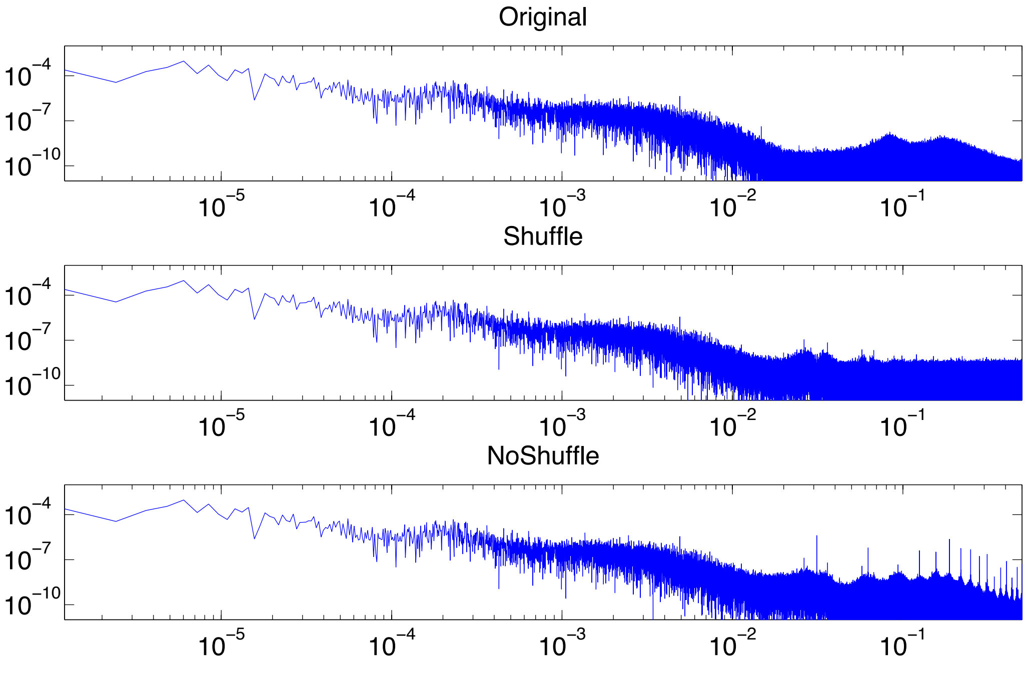


Fig. 5: FFT Spectra of the sample EEG data shown in Fig. 4. When reproducing the data block compressed out, whether or not to shuffle the dictionary data can affect the spectral properties at higher frequencies.

appears earlier in the data. Instead of measuring the similarity of two blocks based on traditional measures such as the Euclidian distance, we use a statistical tool known as Kolmogorov-Smirnov test (KS test). Through a simple design choice, we are able to keep distinctive features in a dataset, while significantly reducing the size needed to keep common data blocks. On two sets of data from very different applications, power grid and neural science, IDEALEM reduces the storage requirement by more than 100-fold, while capturing important features in the data such as voltage sags and current spikes at the same time. In the neural science dataset, it is able to preserve the strong Fourier components and the Fourier components that span multiple data blocks. In both cases, IDEALEM is able to dramatically reduce the storage requirement while preserving features important to the domain scientists, which clearly demonstrate the usefulness of the compression method.

We have a number of tasks planned to extend the IDEALEM compression method. For example, we have mentioned three parameters that affects IDEALEM performance, one immediate plan is to quantify how these parameters affects the compression ratio and other properties. The current version works on data as a sequence, one future plan is to extend this technique to multi-dimensional arrays that are common in scientific applications. In addition, dictionary-based compression methods are known to support a wide range of analysis operations without decompression, we would like to exercise this property to support more efficient operations on the compressed data. Additionally, IDEALEM is not the only way to realize the idea of statistical data reduction, and KS test is not exactly a test for exchangeability. We'd like to explore other options to expand the tools for data reduction.

Acknowledgments

The authors gratefully acknowledge helpful discussions with Kris Bouchard, David Donofrio, Emma Stewart, Sean Peisert, Chuck McParland, Reinhard Gentz, Mahdi Jamei, Ciaran Roberts, and Sebastian Ainslie. 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 work was also supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science, ICT & Future Planning (MSIP) (NRF-2014R1A1A1002662, NRF-2014M2A8A2074096). This research used resources of the National Energy Research Scientific Computing Center.

References

[1] S. Ahern, A. Shoshani, K.-L. Ma, A. Choudhary, T. Critchlow, S. Klasky, V. Pascucci, J. Ahrens, E. W. Bethel, H. Childs, J. Huang, K. Joy, Q. Koziol, G. Lofstead, J. S. Meredith, K. Moreland, G. Ostrouchov, M. Papka, V. Vishwanath, M. Wolf, N. Wright, and K. Wu, "Scientific discovery at the exascale, a report from the doe ascr 2011 workshop on exascale data management, analysis, and visualization," https://science.energy.gov/∼/media/ascr/pdf/program-documents/docs/Exascale-ASCR-Analysis.pdf, 2011.

[2] D. A. Reed and J. Dongarra, "Exascale computing and big data," *Commun. ACM*, vol. 58, no. 7, pp. 56–68, Jun. 2015. [Online]. Available: http://doi.acm.org/10.1145/2699414

[3] K. Sayood, *Introduction to data compression*, 4th ed. Newnes, 2012.

[4] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Inf. Theory*, vol. 23, no. 3, pp. 337–343, May 1977.

[5] S. Di and F. Cappello, "Fast error-bounded lossy HPC data compression with SZ," in *Proc. Int'l Parallel Distrib. Process. (IPDPS '16)*, 2016, pp. 730–739.

[6] P. Lindstrom, "Fixed-rate compressed floating-point arrays," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 2674–2683, December 2014.

[7] M. Burtscher and P. Ratanaworabhan, "FPC: a high-speed compressor for double-precision floating-point data," *IEEE Trans. Comput.*, vol. 58, no. 1, pp. 18–31, January 2009.

[8] J. Iverson, C. Kamath, and G. Karypis, "Fast and effective lossy compression algorithms for scientific datasets," in *Proc. Int'l Conf. Parallel Process. (Euro-Par '12)*, 2012, pp. 843–856.

[9] S. Lakshminarasimhan, N. Shah, S. Ethier, S. Klasky, R. Latham, R. Ross, and N. F. Samatova, "Compressing the incompressible with ISABELA: in-situ reduction of spatio-temporal data," in *Proc. Int'l Conf. Parallel Process. (Euro-Par '11)*, 2011, pp. 366–379.

[10] P. Diaconis, "Recent progress on de finettis notions of exchangeability," *Bayesian statistics*, vol. 3, pp. 111–125, 1988.

[11] J. Choi, K. Hu, and A. Sim, "Relational dynamic Bayesian networks with locally exchangeable measures," Lawrence Berkeley National Laboratory, Tech. Rep. LBNL-6341E, July 2013.

[12] D. Lee, A. Sim, J. Choi, and K. Wu, "Novel data reduction based on statistical similarity," in *Proc. Int'l Conf. Scient. Stat. Database Manag. (SSDBM '16)*, 2016, pp. 21:1–21:12. [Online]. Available: http://doi.acm.org/10.1145/2949689.2949708

[13] A. Sim, D. Lee, K. Wu, and J. Choi, "IDEALEM," November 2016. [Online]. Available: http://datagrid.lbl.gov/idealem

[14] I. E. Richardson, *The H.264 Advanced Video Compression Standard*, 2nd ed. John Wiley and Sons, 2010.

[15] P. Skibiński, S. Grabowski, and S. Deorowicz, "Revisiting dictionary-based compression," *Software: Practice and Experience*, vol. 35, no. 15, pp. 1455–1476, 2005.

[16] E. M. Stewart, S. Kiliccote, C. McParland, C. Roberts, R. Arghandeh, and A. von Meier, "Using micro-synchrophasor data for advanced distribution grid planning and operations analysis," Lawrence Berkeley National Laboratory, Tech. Rep. LBNL-6866E, July 2014.

[17] M. H. Wen, R. Arghandeh, A. von Meier, K. Poolla, and V. O. Li, "Phase identification in distribution networks with micro-synchrophasors," in *Proc. Power & Energy Soc. Gen. Meet. (PES-GM '15)*, 2015, pp. 1–5.

[18] J.-P. Lachaux, N. Axmacher, F. Mormann, E. Halgren, and N. E. Crone, "High-frequency neural activity and human cognition: Past, present and possible future of intracranial EEG research," *Progress in Neurobiology*, vol. 98, no. 3, pp. 279 – 301, 2012, high Frequency Oscillations in Cognition and Epilepsy. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0301008212001062

[19] L. Muller, L. S. Hamilton, E. Edwards, K. E. Bouchard, and E. F. Chang, "Spatial resolution dependence on spectral frequency in human speech cortex electrocorticography," *Journal of Neural Engineering*, vol. 13, no. 5, p. 056013, 2016. [Online]. Available: http://stacks.iop.org/1741-2552/13/i=5/a=056013

[20] P. Lindstrom, "zfp & fpzip: floating point compression," February 2016. [Online]. Available: http://computation.llnl.gov/projects/floating-point-compression