Multidimensional Compression With Pattern Matching

Olivia Del Guercio[1], Rafael Orozco[2], Alexander Sim[3], John Wu[3]
[1]Scripps College,[2]Bucknell University, [3]Lawrence Berkeley National Laboratory

Introduction

Scientific data is usually collected to a greater degree of precision than is significant. Can we take advantage of this feature to reduce storage?
We are working on a dictionary-based compression algorithm that finds statistically similar 1-dimensional data blocks. Here, we consider multidimensional similarity based compression methods as measured by peak signal-to-noise ratios (PSNR) and runtime over different compression levels.

Similarity Measures

- Data points within partitioned time series are assumed to have the statistical property of exchangeability
- Kolmogorov-Smirnov (KS) statistical similarity test can then be performed

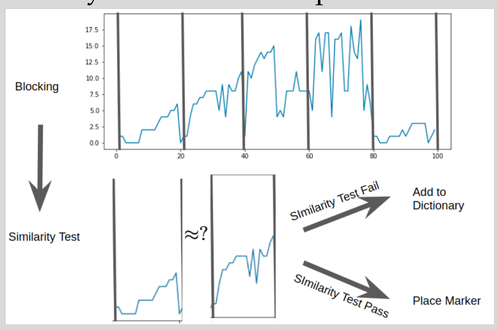


Figure 1: Schematic of a dictionary-based data compression algorithm known as IDEALEM (Implementation of Dynamic Extensible Adaptive Locally Exchangeable Measures)

- The KS test is not multidimensional
- We propose alternative similarity measures: **Dynamic Time Warp** (DTW) and **Minimum Jump Cost** (MJC)
- Consider two time series **x** and **y**

$\mathbf{x} = (4, 4, 3, 2, 4, 0)$
$\mathbf{y} = (7, 5, 6, 4, 3, 4)$

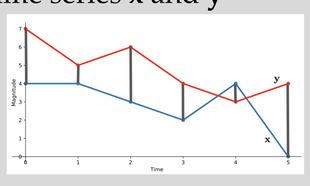


Figure 2: Euclidean distance between time series **x** and **y**

- The most common measure of similarity between time series is the Euclidean distance (Fig. 2) $d_{\mathrm{Euc}}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{M}(x_i - y_i)^2}$

Research Question

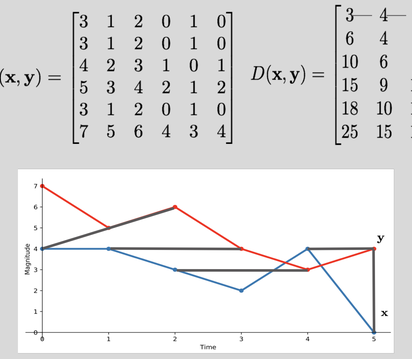
How well can multidimensional similarity based compression work on scientific data?

Dynamic Time Warp

- DTW performs nonlinear "warping" on the sequences where differences in time are not penalized
- For time series of length n, it is necessary to do n^2 computations

$$d_{\mathrm{DTW}}(\mathbf{x}, \mathbf{y}) = D_{M,N}$$
$$D_{i,j} = d_{\mathrm{Euc}}(x_i, y_i) + \min\{D_{i,j-1}, D_{i-1,j}, D_{i-1,j-1}\}$$

$$A(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} 3 & 1 & 2 & 0 & 1 & 0 \\ 3 & 1 & 2 & 0 & 1 & 0 \\ 4 & 2 & 3 & 1 & 0 & 1 \\ 5 & 3 & 4 & 2 & 1 & 2 \\ 3 & 1 & 2 & 0 & 1 & 0 \\ 7 & 5 & 6 & 4 & 3 & 4 \end{bmatrix} \quad D(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} 3 & 4 & 6 & 6 & 7 & 7 \\ 6 & 4 & 6 & 6 & 7 & 7 \\ 10 & 6 & 7 & 7 & 6 & 7 \\ 15 & 9 & 10 & 12 & 7 & 8 \\ 18 & 10 & 11 & 10 & 8 & 7 \\ 25 & 15 & 16 & 14 & 11 & 11 \end{bmatrix}$$



$d_{\mathrm{DTW}} = D_{6,6} = 11$

Figure 3: Visualization of a DTW distance measurement

Minimum Jump Cost

- MJC works by accumulating the cost of jumping forward from one time series data point to the nearest data point in the other time series $d_{\mathrm{MJC}} = \sum c_{\min}^{(i)}$

$$c_{\min}^{(i)} = \min\{c_{t_x}^{t_y\, i}, c_{t_x}^{t_y+1}, c_{t_x}^{t_y+2}, \ldots\}$$

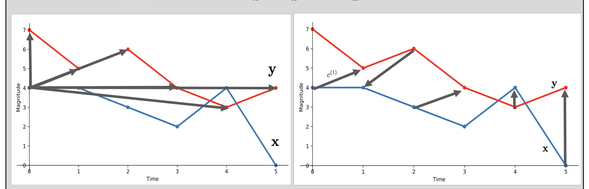


Figure 4: MJC for the first data point (left). Total jumps (right).

- Instead of calculating all n^2 distance values of between **x** and **y**, only the distance between points of index greater than the recursive starting point are calculated
- Expected to reduce runtime

Workflow

Debug existing code

Understand algorithms to implement

Empirical comparison of MJC and DTW

Comparison with interpolation-based algorithms

Comparison Metrics

- Two common performance metrics: compression ratio (CR) and peak signal-to-noise ratio (PSNR)
- CR is the ratio between uncompressed and compressed file sizes

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

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

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

DTW vs. MJC

- MJC has greater efficiency, for only a slight increase in error

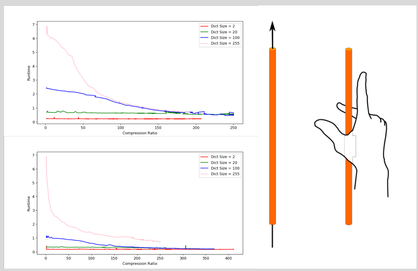


Figure 5: CR vs. Runtime for LBNL Power grid Dataset for DTW (top) and MJC (bot). Illustration of Ampere's Law (left)

100 CR

Dictionary Size		PSNR	Runtime	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: Dictionary size comparison for 100 CR

MJC has lower error at larger dictionary size

DTW is 2x more computationally expensive

Results

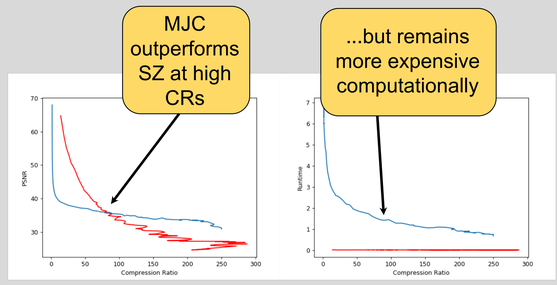


Figure 6: CR vs. PSNR (left) and Runtime (right) for SZ (red) and MJC (blue).

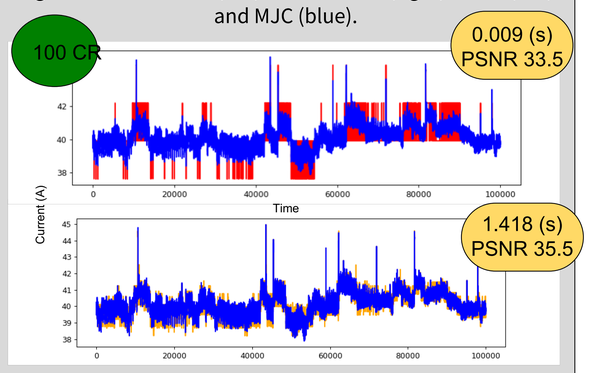


Figure 7: MJC reconstruction performance (bot) and SZ (top) for power grid data.

- Greater PSNR comes at the cost of efficiency
- Desirable features in reconstructed data is often dependent of the set
- SZ relies on nearby points to perform reconstruction, leading to inaccurate decompression in highly variable data

Key Result

Reduce the storage requirements over 100-fold while maintaining relatively low errors in PSNR.

Acknowledgments

This work was supported by the Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231, and also supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists (WDTS) under the Science Undergraduate Laboratory Internship (SULI) program.