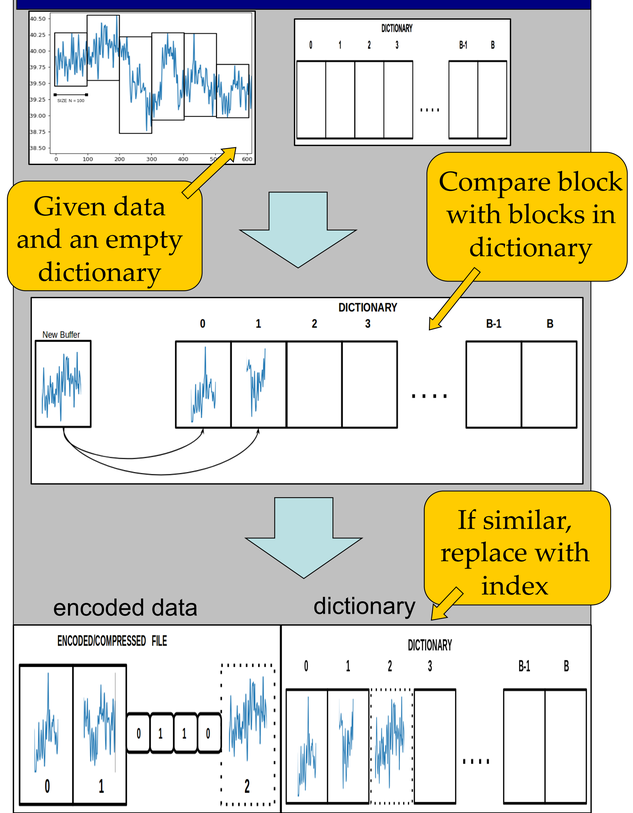
Re-engineering the IDEALEM data compression software

Kunal Agarwal[1], Alex Sim[2], John Wu[2]
[1]University of California, Berkeley, [2]Lawrence Berkeley National Laboratory

ABSTRACT

In a largely data driven world, lots of scientific fields are generating large amounts of data, far too much to analyze. Data compression has become more important than ever. IDEALEM is a lossy compression algorithm but preserves the important characteristics in the data. This work helps expand and generalize the software so it can be easily used by a wide variety of scientific groups.

BACKGROUND INFO



- Algorithm uses a similarity measure to compare data blocks with those saved in the dictionary.
- If data is similar, the index of the block from the dictionary. replaces the block in the data.
- Preserves fluctuating data but compresses stagnant data.

RESEARCH QUESTION

How do we expand the functionality of the IDEALEM software to increase the usability for the user?

LIMITATIONS

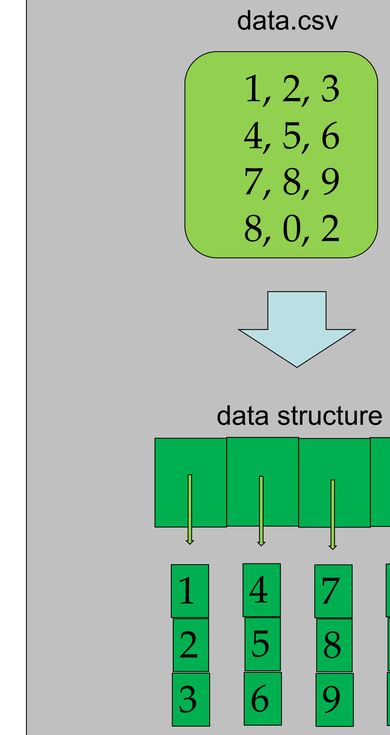
- Previously, software had limitations:
  - Only 1-D data could be inputted
  - Only one similarity measure could be chosen (KS test)
  - Only way to run algorithm was with command line arguments

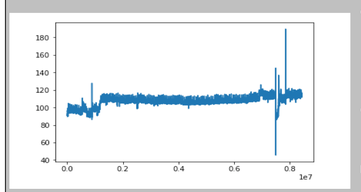
METHODS

- New similarity measures, such as MJC (minimum jumping cost) and DTW (dynamic time warp), are added, allowing the user to pick which one to use.
- API created so user can integrate algorithm into their software.
- The API also allows them to define and use their own similarity measure.
- User can simply call a function in the API that takes in a function pointer, and pass in a pointer to their similarity measure. The software will then use this new function as a similarity measure.
- Tested the API for custom function with user-defined MMD (maximum mean discrepancy) similarity measure.

RESULTS

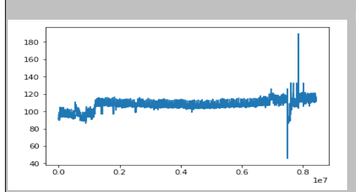
- Data inputted is now stored in this data structure to allow multivariable functionality:





Original data

Encoded data



Recovered data

- Electricity PowerGrid Data
- Size: 133.84 MB

- Parameters:
  - KS Test
  - Block Size: 16
  - Threshold: 0.05
  - History: 254
- Compressed Size: 710.71 KB
- Ratio: 188.32

- Recovered Data after compression
- Very similar to original data

CONCLUSION

- Adding new functionality can allow users for greater autonomy when using the product.
- More independence on how the user wants to use the algorithm and with what data.
- With more simplicity, there is a potential for more users to take advantage of the compression algorithm.

FUTURE WORK

- Using the algorithm to analyze time series data and detect deviations from expected patterns.
- Encoding parameters used to encode data inside the encoded data so that decoding can be done without remembering the parameters used.

ACKNOWLEDGMENTS

This work was prepared in partial fulfillment of the requirements of the Berkeley Lab Undergraduate Research (BLUR) Program, managed by Workforce Development & Education at Berkeley Lab. 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.