

Using Machine Learning to Predict Effective Compression Algorithms for Heterogeneous Datasets

Brandon Alexander Burtchell* and Martin Burtscher

Department of Computer Science

Texas State University

Motivation

- **Heterogeneous** datasets are prevalent in big-data (e.g., IoT¹ and medicine²)
- Data compression is necessary on large datasets
- Using a single compression algorithm on all files is suboptimal
 - Compression algos exploit patterns/redundancies that are unique to particular types of data
- Exhaustively considering many algorithms per file is infeasible

¹Cios and Moore, "Uniqueness of medical data mining".

²Wang, "Heterogeneous Data and Big Data Analytics".

Motivation

- **Heterogeneous** datasets are prevalent in big-data (e.g., IoT¹ and medicine²)
- Data compression is necessary on large datasets
- Using a single compression algorithm on all files is suboptimal
 - Compression algos exploit patterns/redundancies that are unique to particular types of data
- Exhaustively considering many algorithms per file is infeasible

Hypothesis

We can **quickly** predict an **effective** lossless compression algorithm for each file in a heterogeneous dataset.

¹Cios and Moore, "Uniqueness of medical data mining".

²Wang, "Heterogeneous Data and Big Data Analytics".

Highlights

- Approach
 - We call our approach "MLcomp"
 - Offloads computation by **training** a nearest-neighbor (1NN) model off-line
 - Per file, compute **features** to predict a **target** compression algo
- Key findings
 - The compression ratios (CRs) of **simple** compression algos make effective features
 - A few features (4) sufficiently **distinguish** files in a heterogeneous dataset
- Main results
 - We **reduce** a search space of over 100,000 algos to 1 well-performing algo for any input
 - On our evaluation dataset, MLcomp reaches **97.8%** of the CR achieved when exhaustively searching our large library of compression algos

Background: CRUSHER

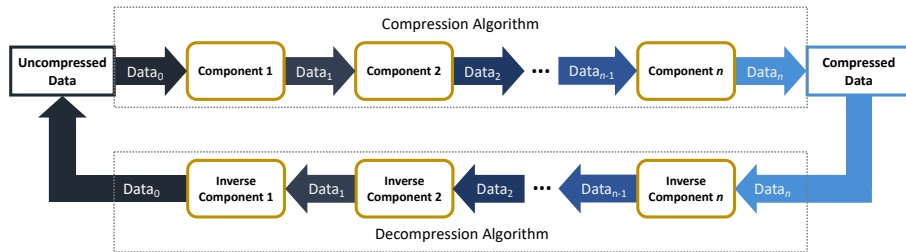


Figure 1: CRUSHER Compression and Decompression Pipeline Flow

- CRUSHER³ generates $56 \times 56 \times 33 = 103,488$ target pipelines
- CRUSHER generates $57 \times 33 = 1,881$ feature pipelines
 - We use sequential feature selection (SFS)⁴ to greedily choose the 4 best features

³Burtscher et al., "Real-Time Synthesis of Compression Algorithms for Scientific Data".

⁴Ferri et al., "Comparative study of techniques for large-scale feature selection".

MLcomp Walkthrough: Setup

- Assumptions

- 12 heterogeneous files to compress: $\{f_0, f_1, f_2, \dots, f_{11}\}$
- 10 CRUSHER components: $\{c_0, c_1, c_2, \dots, c_9\}$

1. Split dataset

- Training: $\{f_0, f_1, f_2, f_3\}$
- Validation: $\{f_4, f_5, f_6, f_7\}$
- Testing: $\{f_8, f_9, f_{10}, f_{11}\}$

2. Generate CRUSHER pipelines

- 10 Features (length 1): $\{c_0, c_1, c_2, \dots, c_9\}$
- 100 Targets (length 2): $\{c_0 c_0, c_0 c_1, c_0 c_2, \dots, c_9 c_9\}$

MLcomp Walkthrough: Setup

- Assumptions

- 12 heterogeneous files to compress: $\{f_0, f_1, f_2, \dots, f_{11}\}$
- 10 CRUSHER components: $\{c_0, c_1, c_2, \dots, c_9\}$

1. Split dataset

- Training: $\{f_0, f_1, f_2, f_3\}$
- Validation: $\{f_4, f_5, f_6, f_7\}$
- Testing: $\{f_8, f_9, f_{10}, f_{11}\}$

2. Generate CRUSHER pipelines

- 10 Features (length 1): $\{c_0, c_1, c_2, \dots, c_9\}$
- 100 Targets (length 2): $\{c_0 c_0, c_0 c_1, c_0 c_2, \dots, c_9 c_9\}$

MLcomp Walkthrough: Setup

- Assumptions

- 12 heterogeneous files to compress: $\{f_0, f_1, f_2, \dots, f_{11}\}$
- 10 CRUSHER components: $\{c_0, c_1, c_2, \dots, c_9\}$

1. Split dataset

- Training: $\{f_0, f_1, f_2, f_3\}$
- Validation: $\{f_4, f_5, f_6, f_7\}$
- Testing: $\{f_8, f_9, f_{10}, f_{11}\}$

2. Generate CRUSHER pipelines

- 10 Features (length 1): $\{c_0, c_1, c_2, \dots, c_9\}$
- 100 Targets (length 2): $\{c_0 c_0, c_0 c_1, c_0 c_2, \dots, c_9 c_9\}$

MLcomp Walkthrough: Training

1. Compute features and identify target pipelines
 - For each **training** file, evaluate the CR of each feature and target pipeline
2. Perform SFS to reduce number of features to $n = 2$
 - Yields c_2 and c_7
3. Train 1NN model with reduced feature vector

Training File	Target Pipeline
f_0	$c_2 c_3$
f_1	$c_4 c_7$
f_2	$c_6 c_1$
f_3	$c_2 c_3$

Table 1: Target Pipeline Lookup

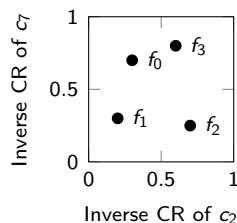


Figure 2: 1NN Feature Space

MLcomp Walkthrough: Prediction

1. Compute feature vector of input file f_8
 - Compute inverse CRs of c_2 and c_7 on f_8
2. Find nearest neighbor (f_2)
3. Compress with neighbor's target pipeline ($c_6 c_1$)

Training File	Target Pipeline
f_0	$c_2 c_3$
f_1	$c_4 c_7$
f_2	$c_6 c_1$
f_3	$c_2 c_3$

Table 1: Target Pipeline Lookup

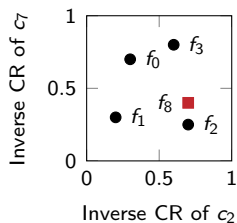


Figure 2: 1NN Feature Space

Evaluation Methodology

- Data is from THEMIS-B satellite⁵
 - 27 distinct data **packet types** sent to Earth daily
 - THEMIS-B assigns compressors according to packet type
- Dataset splits
 - Training: January and February 2013 (1,406 files)
 - Validation: March 2013 (775 files)
 - Testing: All data packets from 2014 (8,916 files)
- Final MLcomp model stats
 - 4 feature pipelines selected from 1,881 (length 2)
 - 90 target pipelines identified from 103,488 (length 3)

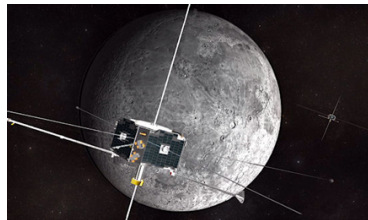


Figure 3: THEMIS Satellite Orbiting the Moon⁶

⁵Angelopoulos, "The THEMIS Mission".

⁶<https://sservi.nasa.gov/wp-content/uploads/drupal/themas1.jpg>

Evaluation Methodology

- Data is from THEMIS-B satellite⁵
 - 27 distinct data **packet types** sent to Earth daily
 - THEMIS-B assigns compressors according to packet type
- Dataset splits
 - Training: January and February 2013 (1,406 files)
 - Validation: March 2013 (775 files)
 - Testing: All data packets from 2014 (8,916 files)
- Final MLcomp model stats
 - 4 feature pipelines selected from 1,881 (length 2)
 - 90 target pipelines identified from 103,488 (length 3)

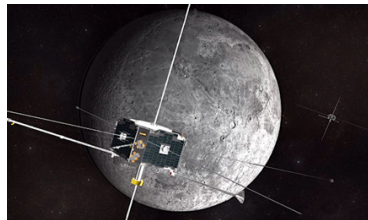


Figure 3: THEMIS Satellite Orbiting the Moon⁶

⁵Angelopoulos, "The THEMIS Mission".

⁶<https://sservi.nasa.gov/wp-content/uploads/drupal/themas1.jpg>

Evaluation Methodology

- Data is from THEMIS-B satellite⁵
 - 27 distinct data **packet types** sent to Earth daily
 - THEMIS-B assigns compressors according to packet type
- Dataset splits
 - Training: January and February 2013 (1,406 files)
 - Validation: March 2013 (775 files)
 - Testing: All data packets from 2014 (8,916 files)
- Final MLcomp model stats
 - 4 feature pipelines selected from 1,881 (length 2)
 - 90 target pipelines identified from 103,488 (length 3)

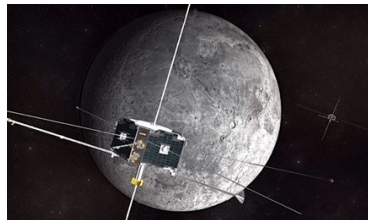


Figure 3: THEMIS Satellite Orbiting the Moon⁶

⁵Angelopoulos, "The THEMIS Mission".

⁶<https://sservi.nasa.gov/wp-content/uploads/drupal/themas1.jpg>

Results: Compression Ratio

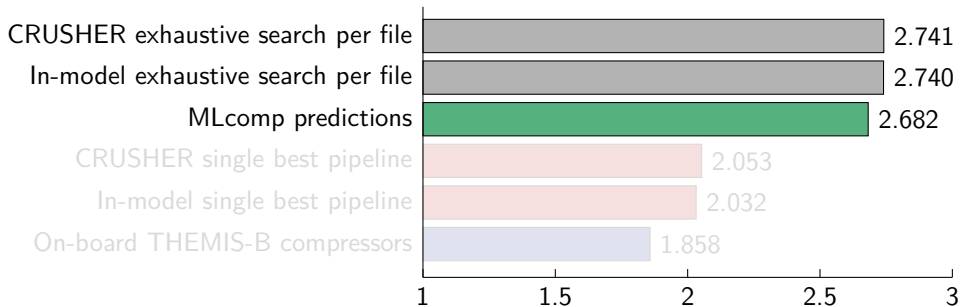


Figure 4: Geometric-mean Compression Ratio of MLcomp and Baselines

- MLcomp nearly achieves the upper bounds (exhaustive searches)
- Compressing with a single pipeline (even the best!) is suboptimal
- MLcomp surpasses THEMIS-B despite withholding the packet type label

Results: Compression Ratio

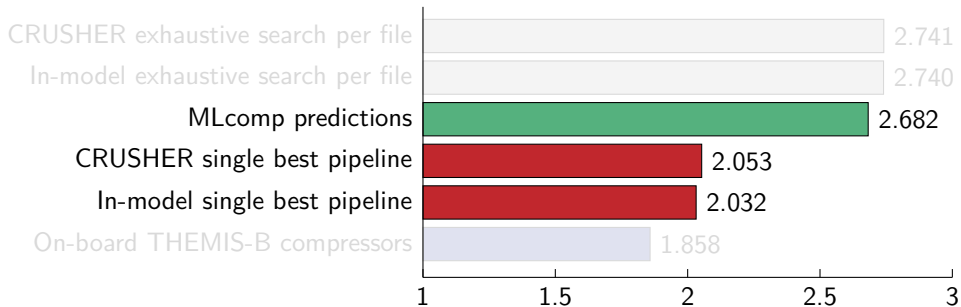


Figure 4: Geometric-mean Compression Ratio of MLcomp and Baselines

- MLcomp nearly achieves the upper bounds (exhaustive searches)
- Compressing with a single pipeline (even the best!) is suboptimal
- MLcomp surpasses THEMIS-B despite withholding the packet type label

Results: Compression Ratio

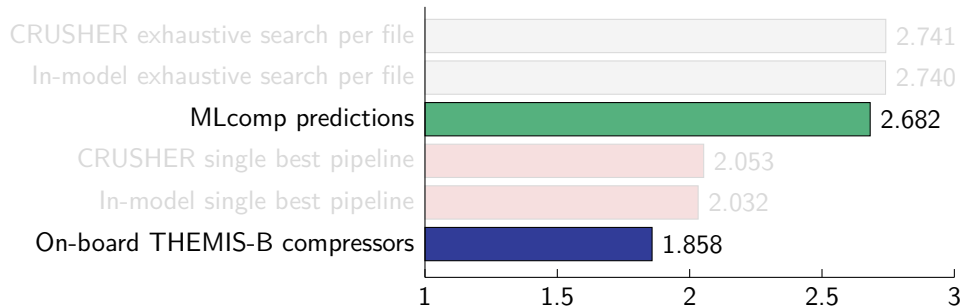


Figure 4: Geometric-mean Compression Ratio of MLcomp and Baselines

- MLcomp nearly achieves the upper bounds (exhaustive searches)
- Compressing with a single pipeline (even the best!) is suboptimal
- MLcomp surpasses THEMIS-B despite withholding the packet type label

Results: Correlation between Packet Type and Predicted Pipeline

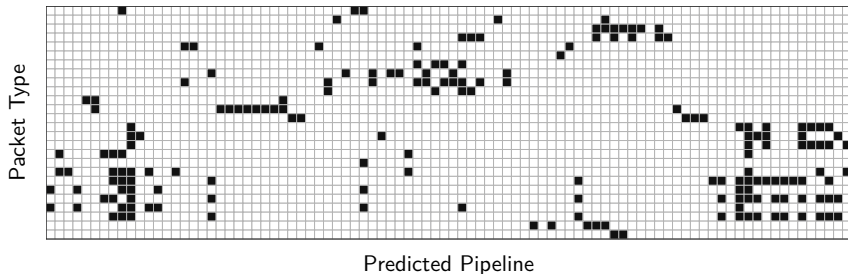


Figure 5: Correlation between Packet Type and Predicted Compression Pipeline

- Discreteness exhibits MLcomp's **lack of bias** towards a few pipelines
- Some packet types have similar sets of predicted pipelines
 - Likely collected by the same instrument in different modes

Results: Comparison with THEMIS-B Compressors (449)

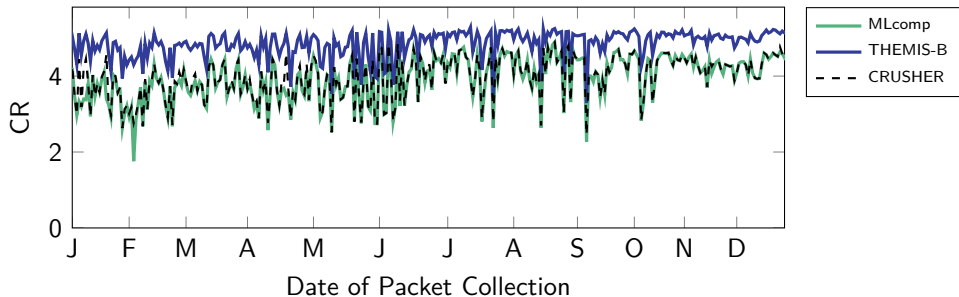


Figure 6: Compression Ratio of Packet Type 449 across Test Set

- Packet type 449: THEMIS-B beats MLcomp by highest factor ($1.2\times$)
 - Due to limitations of CRUSHER's library of compressors, not MLcomp's prediction method

Results: Comparison with THEMIS-B Compressors (45f)

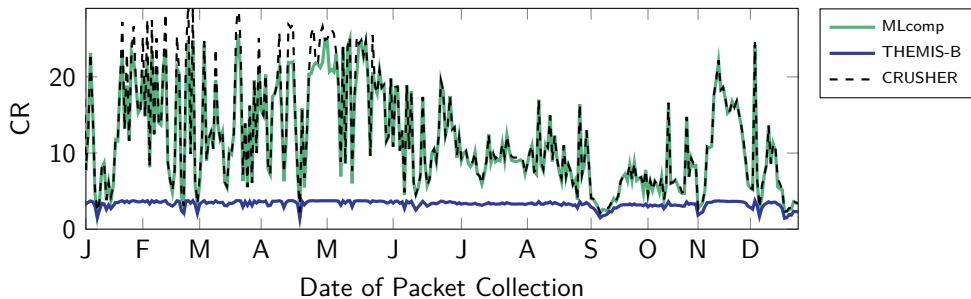


Figure 7: Compression Ratio of Packet Type 45f across Test Set

- Packet type 45f: MLcomp beats THEMIS-B by highest factor ($3.0\times$)
 - MLcomp predicts 15 distinct pipelines (second-highest) for packet type 45f
 - **Adapts** to heterogeneity within packet type

Summary & Conclusion

- **Simple** ML approach can predict effective compression algos for heterogeneous datasets
 - Using a single algorithm on all files results in poor CRs
 - Exhaustively searching for the best algo per file is too slow
- Training a model offloads computation, so prediction is relatively fast
- MLcomp yields **near-optimal** CR on 8,916 unseen heterogeneous packets
- We hope this inspires others to explore ML to improve data compression

Further questions?

burtchell@txstate.edu