Powerstrip: High-Performance Compression for Energy Data

John R. Ward
Okta, Inc.
jack.ward@okta.com

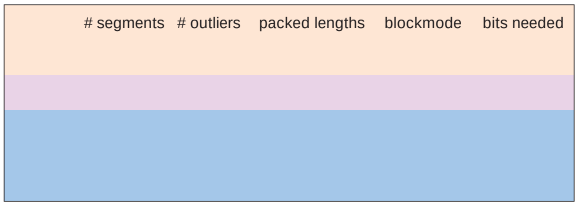
Contributions.

This problem is NP-hard, so various efficient approximations exist. Matching Pursuit [29] is a simple greedy algorithm: for every iteration, simply choose the atom that best matches the difference between the signal and the current approximation. More recent methods include Orthogonal Matching Pursuit [13], which updates all coefficients on every iteration, and Basis Pursuit [10], which converts the optimization into a linear programming problem.

A key challenge in dictionary coding is building dictionaries that effectively approximate the signal. A well-known dictionary learning algorithm is K-SVD [1], which alternates between coding and dictionary improvement using k-means clustering and Orthogonal

reconstructing a time series of *(time, watts)* pairs requires either a consistent recording interval and two pieces of metadata (start time and interval size) or external storage of index values.

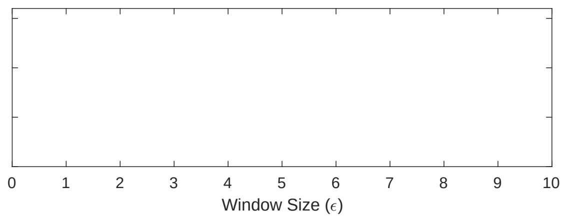
For a given data sequence input, Powerstrip begins by dividing



segments
outliers
packed lengths
blockmode
bits needed

Name	Type	Description
libdeflate [6]	Lossless	LZ77 dictionary coder, as in `gzip`
LZMA [31]	Lossless	Extension of LZ77
Zstd [11]	Lossless	LZ77+asymmetric numeral systems
Simple8b [2]	Lossless	Integer coder
FastPFOR [28]	Lossless	Optimized integer coder
Sprintz [7]	Lossless	IoT integer time series compressor
Uniform quant.	Lossy	Quantization
Gaussian quant.	Lossy	Quantization
PAA	Lossy	Downsampling
K-SVD [34]	Lossy	Sparse dictionary learning

Table 2: The reference compression algorithms considered.


0
1
2
3
4
5
6
7
8
9
10
Window Size (ϵ)