Measuring Swampiness: Quantifying Chaos in Large Heterogeneous Data Repositories

Luann Jung[1], Brendan Whitaker[2], Kyle Chard[3] (advisor), Aaron Elmore[3] (advisor)
[1]Massachusetts Institute of Technology, [2]Ohio State University, [3]University of Chicago

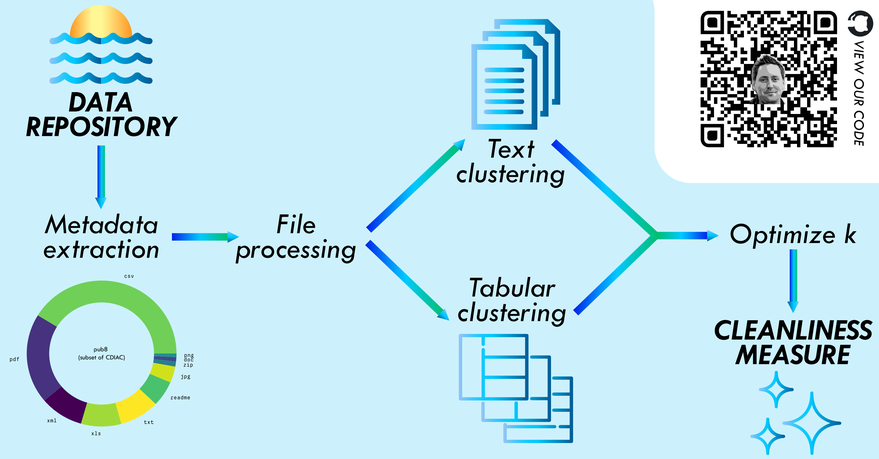
Introduction

- We explore a novel clustering-based approach for organizing data "swamps" by automatically identifying latent content similarities between files.
- We developed a parallel pipeline that crawls large filesystems, collects key information regarding data composition and distribution, and then clusters files automatically based on extracted content and metadata.
- To evaluate our methods we propose a novel method for quantifying the organization of a filesystem using a score based on the directory composition of the clusters.

Clustering pipeline

- **Metadata extraction**
  Extract metadata regarding filetypes within the dataset
- **File processing**
  Unstructured text data:
  - filetypes for text (txt, pdf, doc, html) are converted to txt for easy processing
  - text data are tokenized, stemmed, counted, and vectorized
  - TFIDF matrix computed

  Structured tabular data:
  - filetypes for tabular data (csv, tsv, xls) are converted to csv for easy processing
  - schema extraction extracts headers for the data
  - pairwise Jaccard distance matrix computed
- **Clustering**
  Text data: k-means and the faster MiniBatch k-means
  Tabular data: agglomerative hierarchical clustering
- **Optimizing k**
  Cluster over a user-specified range of k values
  Select the k value that yields the highest frequency drop score
- **Cleanliness score**
  Report frequency drop score associated with best k value



Frequency drop score

Clusters

$$C = \{C_1, ..., C_k\} \in \Gamma.$$

Frequency of a directory in a cluster

$$f_i(D) = |\{x \in D : x \in C_i\}|.$$

Order directories by descending frequency

$$f_i(D_{j_2}) \leq f_i(D_{j_1}) \text{ if } j_1 < j_2.$$

Set of differences in directory frequency

$$W_i = \{f_i(D_{j+1}) - f_i(D_j) : 1 \leq j \leq m-1\}.$$

Index of largest drop in frequency

$$\delta_i = \max\left\{\underset{j}{\operatorname{argmax}}\{f_i(D_{j+1}) - f_i(D_j) : 1 \leq j \leq m-1\}\right\}.$$

Head and tail of the frequency distribution

$$H_i = \{D_j \in L_i : 1 \leq j \leq \delta_i\},$$
$$T_i = \{D_j \in L_i : \delta_i < j \leq m\}.$$

Logarithm-like function defined for base of 1

$$\sigma(a,b) = \begin{cases} \log_a b & \text{if } a > 1 \\ 0 & \text{if } a = 1 \end{cases}.$$

Frequency drop score for a cluster

$$\text{drop}(C_i) = \begin{cases} \frac{1-\sigma(m-1,|H_i|)}{|C_i|} \sum_{D_j \in H_i} |D_j| & \text{if } m > 1 \\ 1 & \text{otherwise} \end{cases}.$$

Sets and elements

Files:	x
Directories:	D_j
Dataset:	A
Clusters:	C_i
Set of directories of C_i :	L_i
Clustering of A :	C
Set of all clusterings of A :	Γ
Set of all directories of A :	L

Quantities

Number of clusters:	k
Number of directories in L_i :	m

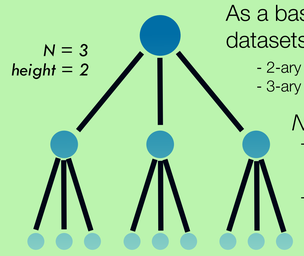
Functions

$$f_i : L_i \to \mathbb{N}$$
$$\sigma : \mathbb{N} \times \mathbb{R}_{>0} \to \mathbb{R}$$
$$\text{drop} : C \to [0,1]$$
$$\mathcal{S} : \Gamma \to [0,1]$$

Frequency drop score

$$\mathcal{S}(C) = \frac{|C_i|}{|A|} \sum_{i=1}^{k} \text{drop}(C_i).$$

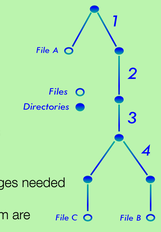
Evaluation on synthetic data



As a baseline for comparison, we generated 4 "perfect" datasets based on *N*-ary trees:

- 2-ary 5-height (expect 32 clusters)
- 3-ary 3-height (expect 27 clusters)
- 6-ary 2-height (expect 36 clusters)
- 40-ary 1-height (expect 40 clusters)

N-ary trees
- root node/directory has *N* children
  - each subsequent node has *N* children
  - extended to some height *h*
- each leaf node is named after some word *w*
  - each leaf node contains 20 files
  - each file contains *w* repeated 100 times

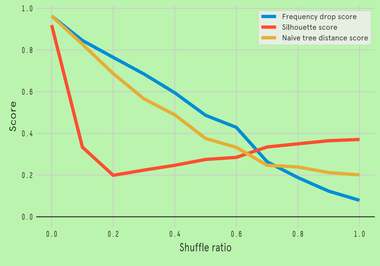


Tree distance
- calculated as the number of directory changes needed until two files are in the same directory
- measures how far apart files in the filesystem are

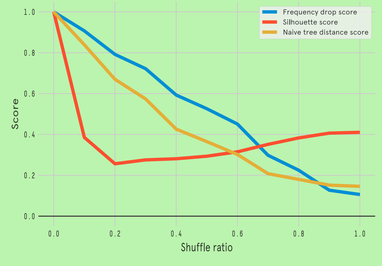
Comparison of scores

Baselines:
- Naive tree distance between files in the filesystem (cohesion):
- Silhouette score calculated using file system tree distance (cohesion and separation)

Cleanliness scores
3-ary tree with height 3.

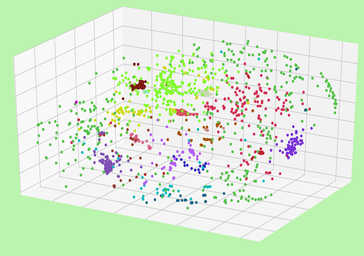


Cleanliness scores
6-ary tree with height 2.

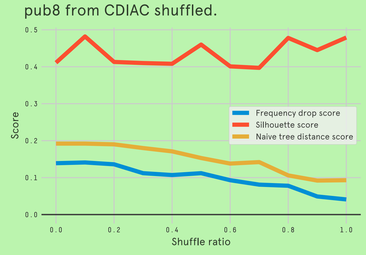


Evaluation on real data

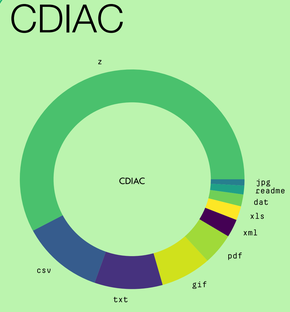
PUB8

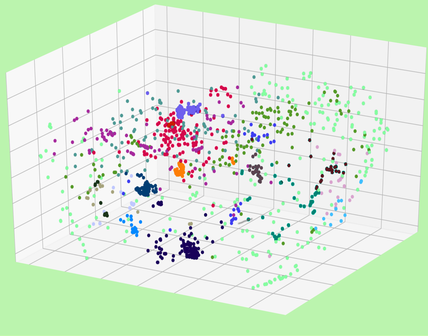


Cleanliness scores
pub8 from CDIAC shuffled.



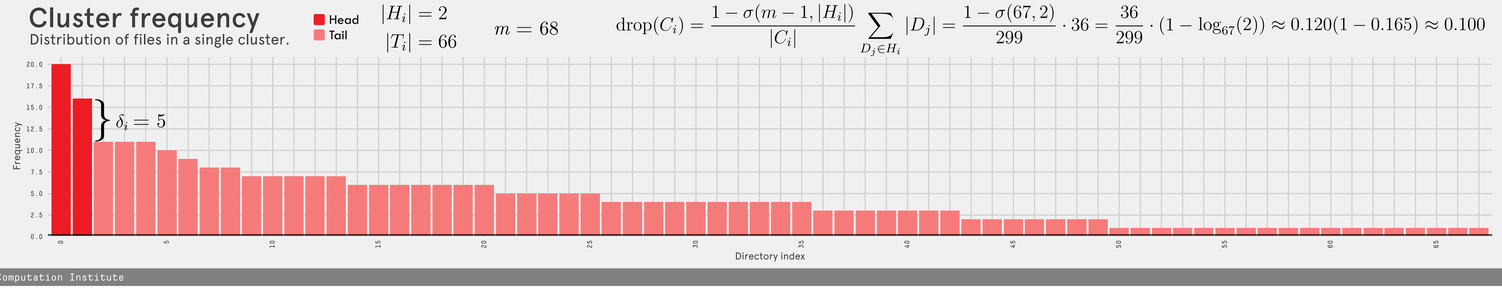
CDIAC





Cluster frequency

Distribution of files in a single cluster.



$$\text{drop}(C_i) = \frac{1-\sigma(m-1,|H_i|)}{|C_i|} \sum_{D_j \in H_i} |D_j| = \frac{1-\sigma(67,2)}{299} \cdot 36 = \frac{36}{299} \cdot (1 - \log_{67}(2)) \approx 0.120(1-0.165) \approx 0.100$$

References

[1] Paul Beckman, Tyler J Skluzacek, Kyle Chard, and Ian Foster. 2017. Skluma:A statistical learning pipeline for taming unkempt data repositories. In *29th International Conference on Scientific and Statistical Database Management*. 41.

[2] Will Brackenbury, Rui Liu, Mainack Mondal, Aaron J. Elmore, Blase Ur, KyleChard, and Michael J. Franklin. 2018. Draining the Data Swamp: A Similarity-based Approach. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics (HILDA'18)*. ACM, New York, NY, USA, Article 13, 7 pages. https://doi.org/10.1145/3209900.3209911

Acknowledgements

Many thanks to Will Brackenbury and Tyler Skluzacek for their help during the brainstorming and editing processes. This work was supported by Chameleon Cloud, Globus, and NSF REU 1757964 BigDataX: From theory to practice in Big Data computing at eXtreme scales.