Skluma: A Statistical Learning Pipeline for Taming Unkempt Data Repositories

Conference Paper · June 2017

DOI: 10.1145/3085504.3091116

4 authors, including:

Tyler J. Skluzacek
University of Chicago
2 PUBLICATIONS **1** CITATION
SEE PROFILE

Kyle Chard
University of Chicago
72 PUBLICATIONS **683** CITATIONS
SEE PROFILE

Ian Foster
University of Chicago
914 PUBLICATIONS **81,304** CITATIONS
SEE PROFILE

Skluma: A Statistical Learning Pipeline for Taming Unkempt Data Repositories

Paul Beckman, Tyler J. Skluzacek, Kyle Chard, and Ian Foster
Computation Institute
University of Chicago and Argonne National Laboratory
Chicago, IL 60637
{pbeckman,skluzacek,chard}@uchicago.edu,foster@anl.gov

ABSTRACT

Scientists' capacity to make use of existing data is predicated on their ability to find and understand those data. While significant progress has been made with respect to data publication, and indeed one can point to a number of well organized and highly utilized data repositories, there remain many such repositories in which archived data are poorly described and thus impossible to use. We present Skluma—an automated system designed to process vast amounts of data and extract deeply embedded metadata, latent topics, relationships between data, and contextual metadata derived from related documents. We show that Skluma can be used to organize and index a large climate data collection that totals more than 500GB of data in over a half-million files.

CCS CONCEPTS

• **Information systems → Data cleaning**; **Mediators and data integration**;

KEYWORDS

data wrangling, statistical learning, metadata extraction, data integration

ACM Reference format:
Paul Beckman, Tyler J. Skluzacek, Kyle Chard, and Ian Foster. 2017. Skluma: A Statistical Learning Pipeline for Taming Unkempt Data Repositories. In *Proceedings of SSDBM '17, Chicago, IL, USA, June 27-29, 2017,* 4 pages.
https://doi.org/http://dx.doi.org/10.1145/3085504.3091116

1 INTRODUCTION

Meaningless file names. Limited documentation. Unlabeled columns. Numerically encoded null values. Multifarious file extensions. Scientists live this nightmare daily as they seek to discover and use publicly available data stored in heterogeneous data repositories. As the rate of data production explodes (e.g., due to higher resolution instruments and massive sensor networks), clear, uniform documentation and organization of data are often neglected. Many efforts have focused on standardizing data naming and organization models within and across research groups [17, 20]. We, and others, have developed workflow-oriented data publication systems that impose requirements on organization and metadata [7]. While there are clear success stories with respect to structured data repositories [4, 9], repositories often become dumping grounds for poorly described data [6]. We postulate that new methods based in statistical learning are needed to make sense of the vast amounts of data already published to existing repositories. To this end, we propose an automated pipeline (Skluma) and associated models and methods that allows us to process and classify disorderly data, striving to provide the metadata necessary to enable complex querying of previously incomprehensible scientific data repositories.

Skluma is organized around a three-stage pipeline: *crawl* heterogeneous data repositories, *extract* metadata on each file's content, and *contextualize* data in order to enrich, augment, and improve the accuracy of existing metadata. Throughout these stages, Skluma accumulates and refines metadata through a number of information extraction and statistical learning models.

To illustrate the value of Skluma we have used it to organize and index the contents of the United States Department of Energy's Carbon Dioxide Information Analysis Center (CDIAC) data store [19]. This filesystem-based repository contains over a half-million files of diverse types, structures, and sizes, distributed among twelve grab-bag 'pub' top-level directories. Many files are compressed, named according to undocumented systems, organized in arbitrary hierarchies, and stored in nonstandard formats. Furthermore, there are duplicate files both within and between directories, and scientifically useless files (e.g., Windows Installers, shortcuts, empty zipped directories). CDIAC contains more than 150 different file extensions, making the implementation of type- or format-specific metadata extractors infeasible. **Figure 1** shows the distribution of file types. CDIAC is illustrative of a common problem in science: while researchers may work hard to address problems of verifiability and reproducibility, these considerations are easily obscured or lost by publishing disorganized and undocumented data. Skluma works to reclaim this missing context and restore the usefulness of disarrayed repositories.

The remainder of the paper is organized as follows: Section 2 provides a brief overview of past methods for database and repository cleaning. Section 3 outlines our pipeline and presents test results. Section 4 describes our planned demonstration of Skluma. Section 5 outlines future work and considerations. Finally, Section 6 provides a full analysis of Skluma.

SSDBM '17, June 27-29, 2017, Chicago, IL, USA

ACM ISBN 978-1-4503-5282-6/17/06…$15.00
https://doi.org/http://dx.doi.org/10.1145/3085504.3091116

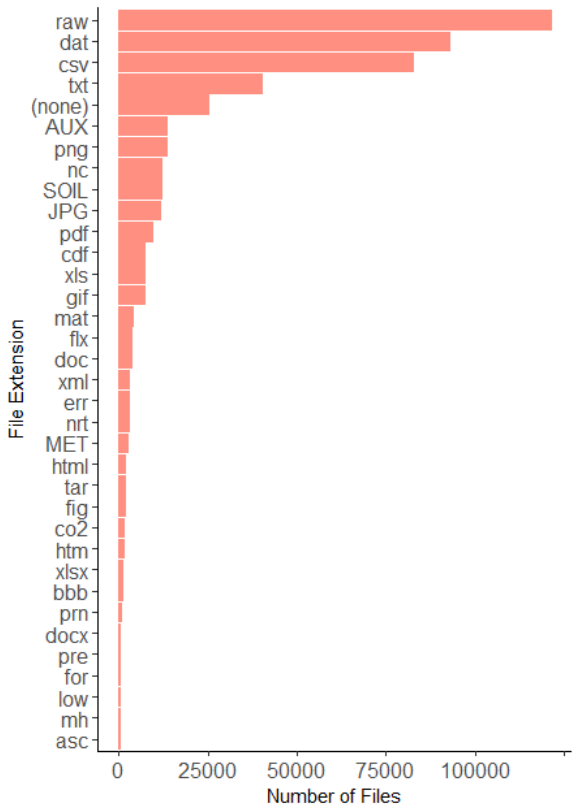

Figure 1: CDIAC file extension distribution: Counts for the 35 most common file extensions in CDIAC.

2 RELATED WORK

Skluma follows a long line of attempts to organize and gain insight into highly disorganized data. Related work on geospatial attribute recognition has paired simple rule-based analysis with the use of support vector machines (SVMs) in order to predict attributes in a broad range of geospatial datasets [2, 14]. Current methods focus on well-structured data with higher consistency than data sources like CDIAC.

Skluma is also not the first to provide a scalable solution to collect raw datasets and extract metadata from them. Pioneering research on data lakes has developed methods for extracting standard metadata from nonstandard file types and formats [18]. Recently the data lake has been adapted to standardize and extract metadata from strictly-geospatial datasets [16]. Normally data lakes have some sort of institution-specific target for which they are optimized, whether they primarily input transactional, scientific, or networked data. Skluma is optimized for data without any standardization guarantees, providing information on related, relevant data and their attributes.

Finally, Skluma supplements existing work that cleans and labels data using context features. Data Civilizer [10] accounts for proximate files in data warehouses by building and interpreting linkage graphs. Others have used context as a means to determine how certain metadata might be leveraged to optimize performance in a specific application or API [15]. Skluma collects and analyzes context metadata in order to allow research scientists to find, query, and download related datasets that aid their scientific efforts.

3 PIPELINE

Skluma implements a three-stage pipeline: (1) crawling, (2) metadata extraction, and (3) contextualization.

3.1 Crawling

Skluma's first task is to catalog the data in order to understand its organization and scope. While crawling, Skluma extracts general file-level metadata, such as file name, path, size, a checksum, extension and MIME type [13] for each file. Given the wide variety of data access protocols (e.g., HTTP, FTP, GridFTP) offered by data repositories, Skluma is designed with a modular crawling architecture in which different crawler implementations can be used. In the case of CDIAC, we used FTP and HTTP crawlers. As a result of the crawling phase Skluma creates a JSON document that stores basic system metadata about each file. This JSON file is stored, modified, and appended to throughout the Skluma pipeline.

In order to extract metadata from within files, Skluma requires a mutable file system to execute decompression if necessary and to store the resulting files. As data repositories often do not provide such resources, the Skluma crawler mirrors data temporarily to a high-performance storage environment, where metadata extraction can be performed. In the CDIAC scenario, we use Globus [12] to transfer all CDIAC data temporarily to Petrel [1], a 1 PB storage system housed at Argonne National Laboratory.

3.2 Metadata Extraction

The second pipeline phase iterates over all files discovered during the crawling phase and extracts metadata from each file based on its content. We leverage a suite of modular extraction tools to obtain general, file, and domain-specific metadata. As these extractors can require significant compute resources, we use Jetstream as a scalable platform on which to execute arbitrary extraction tools on the data. Petrel serves as a performant storage system from which we can rapidly access data for processing on Jetstream.

The end goal of extracted metadata is to facilitate queries over a repository with field-specific predicates (e.g., "return all CDIAC files with temperature values greater than 20 C"). To this end, Skluma's metadata extractors address two main types of data: *containerized* and *column-formatted*. Containerized data formats like NetCDF already include much of the metadata necessary for query construction accessible in standard formats and via standard interfaces. In this case, Skluma simply reformats this information and copies it into the metadata file. For any file that can be parsed as column-formatted, we calculate the min, max, and average for numerical columns. In addition, we collect all available headers.

In order for these aggregates to have any meaning, however, encoded null values must be detected and (typically) skipped. Otherwise, to use a CDIAC example, we may find that a scientist records -999 to indicate that no temperature measurement was taken, leading to a file with an average temperature of -437 C. Thus Skluma employs a supervised learning model to infer null values and exclude them from aggregate calculations. We use a k-nearest neighbor

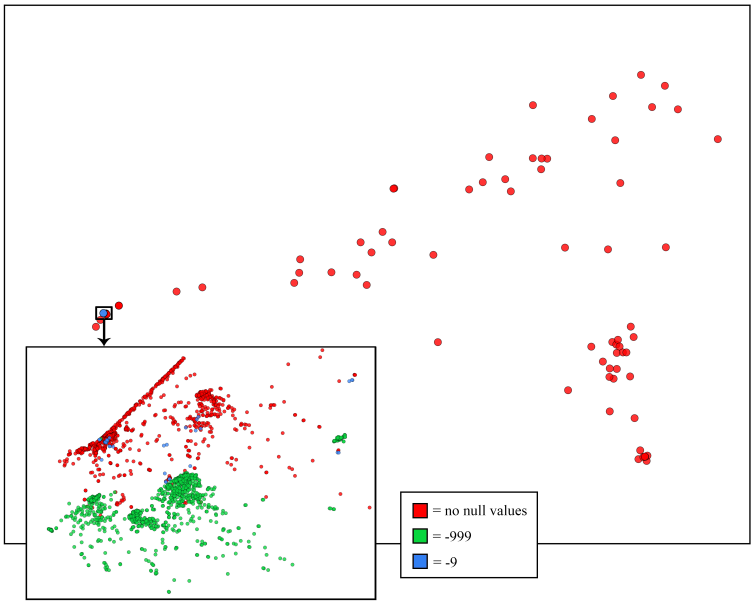

Figure 2: PCA visualization of null values: Cutout shows dense region of the feature space at 2500x zoom.

classification algorithm, using the average, the three largest values and their differences, and the three smallest values and their differences as features for our model. By taking a classification rather than regression-based approach, Skluma selects from a preset list of null values, which avoids discounting real experimental outliers recorded in the data itself. **Figure 2** provides a PCA visualization of the clustering of null values in the feature space.

When trained and tested by cross-validation on a labeled test set of 4682 columns from 335 unique files, our model achieved accuracy 0.991, precision 0.989, and recall 0.961, where precision and recall are calculated by macro-averaging over classifiers.

At this point in the pipeline the accuracy of the aggregate values has been improved to better reflect the existing data. However, if the column header is not provided in the file itself, these values provide very little information that can be used for query or discovery. This is a common occurrence in CDIAC data. Files often contain only numerical values, whereas column name information is in a separate free-text README file in a nearby directory. The problem of associating unlabelled data columns with headers is addressed by the third portion of the Skluma pipeline.

3.3 Contextualization

The relationships and similarities between files within and outside a data repository can provide valuable information regarding the nature of these files. Specifically, topic labels on free-text documents can serve as valuable context to describe nearby undocumented data. Skluma employs a topic mixture model based on Latent Dirichlet Allocation [5]. Our model is made up of three steps. First, we train our model on Web of Science (WoS) abstracts. Next, we use this model to generate the topic distribution of each free-text README or documentation file in the data repository, which is the finite mixture over an underlying set of topics derived from the model. Finally, we model all data files in the repository as themselves finite mixtures of the topic distributions of the surrounding labelled files. We calculate the topic mixture of a given file as a linear combination $w_1 d_1 + w_2 d_2 + \ldots + w_n d_n$ of the topic distributions $d_1, \ldots, d_n$ of all nearby tagged free-text documents within a distance threshold. The weights $w_1, \ldots, w_n$ are inversely proportional to the distance within the repository of the data file to the tagged text document. The distance metric we use is dependent on the type of repository being considered. For directory-structured file systems like CDIAC, we use the number of directory changes that must be done in order to reach one file from another. A simple illustration of this model is shown in **Figure 3**. We then execute these steps by submitting a series of jobs to the Cloud Kotta [3] platform.

This model serves three purposes. Firstly, it adds an additional queryable attribute to the metadata, enabling searches by probable topic. Secondly, it can act as a basis for selecting specialized metadata extraction tools for classified files. Finally, it may be used as a feature for a statistical learning model used to predict missing column headers. We discuss this final prospect in Section 5.

4 DEMO

The demonstration of Skluma involves executing our pipeline on a small subset of CDIAC. Specifically, we will demonstrate inference of attributes and null-values, tagging files with topics via analysis of their proximate READMEs, and extracting metadata such that the files are queryable by both their content and topic-context. Furthermore, we will query over Skluma's resulting metadata by using a simple, web-based search GUI built atop ElasticSearch. The

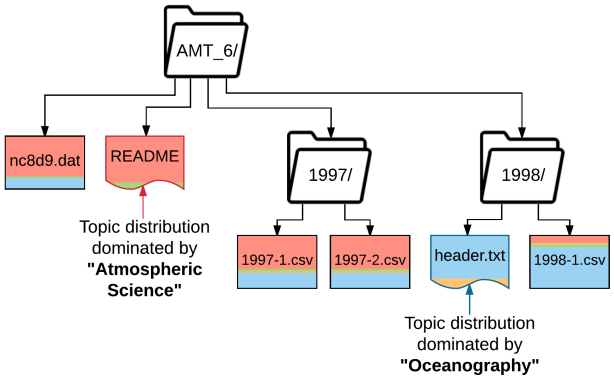

Figure 3: LDA label mixture model: Files are colored according to the relative weight of the surrounding topics; red represents "Atmospheric Science" and blue "Oceanography."

takeaway from the demo should be as follows: despite the repository's disorganized structure and content, Skluma is able to provide informative metadata that can be used to facilitate data discovery.

5 FUTURE WORK

For column-structured data, one important step towards facilitating data discovery is the inference of column headers in unlabelled data. At this juncture, Skluma can produce accurate aggregates by removing null values and provide topic-based context for headerless files. We intend to develop additional statistical learning models that leverage these informative features in conjunction with further natural language processing techniques. These approaches may allow us to use free-text documentation files to predict specific column headers in nearby files, which would greatly increase the amount of previously unsearchable data that can be indexed for querying and discovery.

Beyond column-formatted data, we will augment the metadata we collect from semi-structured and unstructured files. To do so, we have begun exploring variants of other schema-extraction paradigms [8, 11] as an initial step in the pipeline. Providing further insight into unstructured data files will broaden the coverage of Skluma's derived metadata, moving towards more comprehensive data discovery.

6 SUMMARY

Skluma's three-step pipeline supports crawling, metadata extraction, and contextualization, working to provide the metadata necessary for a data querying environment for scientists. We employ a number of statistical learning models in order to determine the characteristics of and relationships between files despite irregularities, missing fields, and haphazard organization. The development and implementation of this class of automated information extraction methods has the potential to greatly expand the quantity of usable scientific data. We continue to expand and refine Skluma in order to better convert unkempt data repositories into clear, searchable resources that can propel novel research and analysis.

REFERENCES

[1] Petrel Data Management and Sharing Pilot. (????). https://www.petrel.alcf.anl.gov. Visited Feb. 28, 2017.
[2] Shilpi Ahuja, Mary Roth, Rashmi Gangadharaiah, Peter Schwarz, and Rafael Bastidas. 2016. Using Machine Learning to Accelerate Data Wrangling. In *Proceedings of the 16th IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, 343–349.
[3] Y. N. Babuji, K. Chard, A. Gerow, and E. Duede. 2016. Cloud Kotta: Enabling secure and scalable data analytics in the cloud. In *Proceedings of the IEEE International Conference on Big Data (Big Data)*. 302–310. https://doi.org/10.1109/BigData.2016.7840616
[4] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. 2000. The Protein Data Bank. *Nucleic Acids Research* 28, 1 (2000), 235. +http://dx.doi.org/10.1093/nar/28.1.235
[5] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.
[6] Carl F Cargill. 2011. Why standardization efforts fail. *Journal of Electronic Publishing* 14, 1 (2011).
[7] Kyle Chard, Jim Pruyne, Ben Blaiszik, Rachana Ananthakrishnan, Steven Tuecke, and Ian Foster. 2015. Globus Data Publication As a Service: Lowering Barriers to Reproducible Science. In *Proceedings of the 2015 IEEE 11th International Conference on e-Science (E-SCIENCE '15)*. IEEE Computer Society, Washington, DC, USA, 401–410. http://dx.doi.org/10.1109/eScience.2015.68
[8] Cloudera. 2014. RecordBreaker. *Cloudera RecordBreaker GitHub Repository* (2014). https://github.com/cloudera/RecordBreaker/tree/master/src
[9] Timothy D. Crum, Ron L. Alberty, and Donald W. Burgess. 1993. Recording, Archiving, and Using WSR-88D Data. *Bulletin of the American Meteorological Society* 74, 4 (1993), 645–653. https://doi.org/10.1175/1520-0477(1993)074<0645:RAAUWD>2.0.CO;2
[10] Dong Deng, Raul Castro Fernandez, Ziawasch Abedjan, Sibo Wang, Michael Stonebraker, Ahmed Elmagarmid, Ihab F Ilyasl, Samuel Madden, Mourad Ouzzani, and Nan Tang. 2017. The Data Civilizer System. In *Proceedings of the 8th Biennial Conference on Innovative Data Systems Research (CIDR)*.
[11] Kathleen Fisher and David Walker. 2011. The PADS project: an overview. In *Proceedings of the 14th International Conference on Database Theory*. ACM, 11–17.
[12] Ian Foster. 2011. Globus Online: Accelerating and democratizing science through cloud-based services. *IEEE Internet Computing* 15, 3 (2011), 70.
[13] N. Freed and N. Borenstein. 1996. *Multipurpose Internet Mail Extensions (MIME)*. RFC 2045. IETF.
[14] Hui Han, C. Lee Giles, Eren Manavoglu, Hongyuan Zha, Zhenyue Zhang, and Edward A. Fox. 2003. Automatic Document Metadata Extraction Using Support Vector Machines. In *Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '03)*. IEEE Computer Society, Washington, DC, USA, 37–48. http://dl.acm.org/citation.cfm?id=827140.827146
[15] Joseph M Hellerstein, Vikram Sreekanti, Joseph E Gonzalez, James Dalton, Akon Dey, Sreyashi Nag, Krishna Ramachandran, Sudhanshu Arora, Arka Bhattacharyya, Shirshanka Das, et al. 2017. Ground: A Data Context Service. In *Proceedings of the 8th Biennial Conference on Innovative Data Systems Research (CIDR)*.
[16] Tyler J. Skluzacek, Kyle Chard, and Ian Foster. 2016. Klimatic: A Virtual Data Lake for Harvesting and Distribution of Geospatial Data. In *Proceedings of the 1st Joint International Workshop on Parallel Data Storage & Data Intensive Scalable Computing Systems (PDSW-DISCS '16)*. IEEE Press, Piscataway, NJ, USA, 31–36. https://doi.org/10.1109/PDSW-DISCS.2016.9
[17] Hiroyasu Sugano, Adrian Bateman, Wayne Carr, Jon Peterson, Shingo Fujimoto, and Graham Klyne. 2004. *Presence information data format (PIDF)*. RFC 3863. IETF.
[18] Ignacio Terrizzano, Peter M Schwarz, Mary Roth, and John E Colino. 2015. Data Wrangling: The Challenging Journey from the Wild to the Lake.. In *Conf. on Innovative Data Systems Research*.
[19] U.S. Dept. of Energy. 2017. Carbon Dioxide Information Analysis Center. (Jan 2017). ftp://cdiac.ornl.gov
[20] John Wieczorek, David Bloom, Robert Guralnick, Stan Blum, Markus Döring, Renato Giovanni, Tim Robertson, and David Vieglais. 2012. Darwin Core: an evolving community-developed biodiversity data standard. *PloS one* 7, 1 (2012), e29715.