Data-Driven Lightweight Encoding Selection

Hao Jiang, Aaron J. Elmore

Department of Computer Science
The University of Chicago

This project explores an automated method to select efficient lightweight encoding schemes for column-based databases. Existing methods either rely on database administrators' experience or use simple rules to make selection. In practice, neither of these methods achieve optimal performance. We propose a method to solve the problem in a systematic data-driven way.

Lightweight Encoding

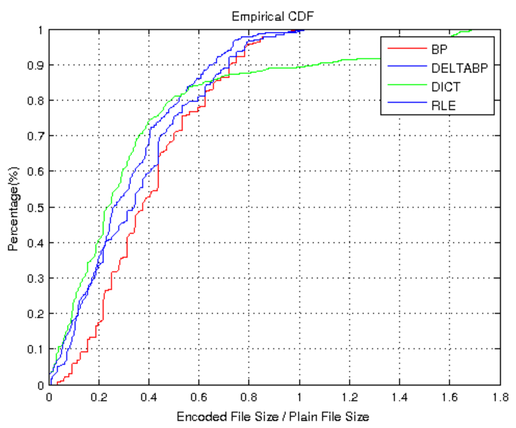
Popular Lightweight encoding schemes includes Run-Length Encoding, Dictionary Encoding, Bit-Packing and Delta Encoding. Comparing to popular compression techniques like gzip and snappy, lightweight encoding schemes have many advantanges such as Encoding Speed, Local Computation and Support to In-Site Query Execution.

Best encoding scheme given a data attributes is determined by many factors including data type and data nature, and there is no simple rule on how to choose it. The system we propose includes the following features:

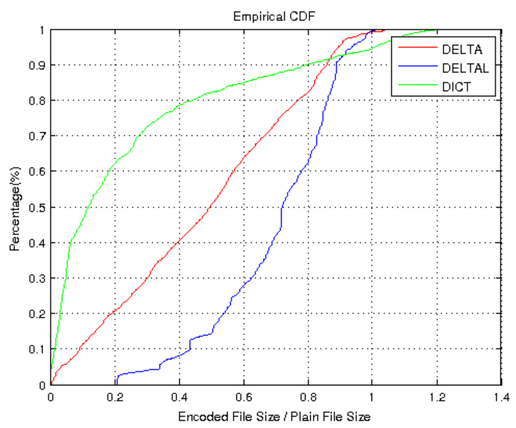
1. Dataset Analysis
2. Pattern Mining
3. Data-Driven Encoding Prediction

Dataset Analysis

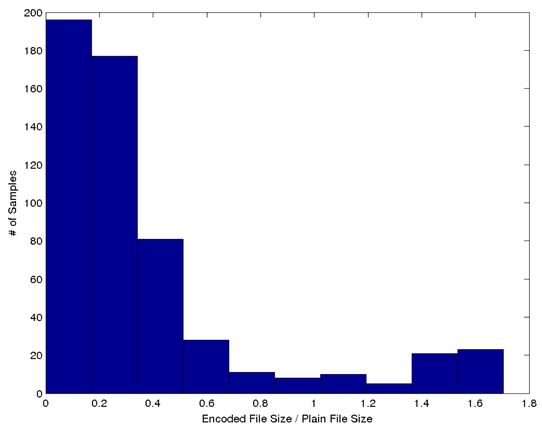
We have created an automated framework to collect datasets, extract columns from them, organize, and persist the records for further analysis. Using this framework, we have collected over 7000 columns from approximately 1200 datasets with a total size of 500G data. These datasets are all from real-world data sources and cover a rich collection of data types (integer, date, address, etc.), with diverse data distributions. We use Apache Parquet's built-in encoders to encode these data columns with different encoding schemes, looking for the one performing best for each column.



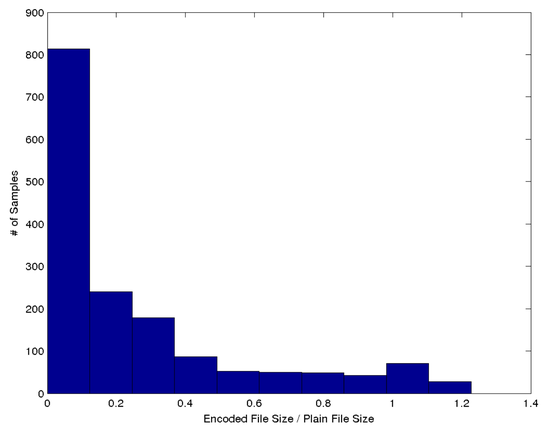
(a) Encoding Compression Ratio for Integer Columns



(b) Encoding Compression Ratio for String Columns



(a) Dictionary Encoding for Integer Columns



(b) Dictionary Encoding for String Columns

Pattern Mining

Data type is crucial to encoding selection. A proper data type determination can greatly reduce space requirement for encoded data. For example, storing a date field in string format requires at least 8 bytes, while storing it in integer format takes no more than 4 bytes. If we further observe the effective data range, this can be further reduced to 23 bits. However, most real-world datasets are semi-structured, in which only part of the data contains valid common structures. Pattern Mining targets at automatically identify and extract these structures, allowing more efficient encoding to be applied on them. In this project, we have developed two methods for Pattern Mining, **Common Sequence** and **Frequent Similar Words**.

Data Driven Encoding Prediction