Final Project – Compressing Datasets for Machine Learning
Submit in pairs. Due: September 10, 2017

The objective of the project is to study *lossless compression for data sets*. Imagine a mobile/IoT device collecting structured data of some kind, which later will (or will not) be used for some *machine-learning* task. Because we do not know exactly how the data will be used, we need to compress it without loss.

1 Preparation

1. Log into the site Kaggle.com (account needed, free sign-up), and look for datasets of the kind collectable by mobile/IoT devices (for example: Mobile location history of 10/2014).
2. After finding 3-4 datasets, look for literature on compression algorithms that can be useful for such data.
3. Have your implementation for Homework 2 handy.

2 Method

1. Your objective is to compress each dataset to a minimal size, such that it can be decompressed back to the original dataset. Dictionaries used for decompression should be counted toward the compressed size.
2. It is recommended that you re-format the dataset to eliminate any meta-data that is not part of the data itself, for example deterministic field names or long field delimiters interspersed with the data.
3. Important: To design the compression algorithm **you are allowed to look at the first 10% of the dataset only**. These 10% will be called from now the *training set.*

3 Tasks

Perform the following tasks on 3 datasets from Kaggle.com.

3.1 Part 1: Huffman Coding

Viewing the dataset as a symbol stream (after re-formatting), use the training set to decide the best input alphabet of the Huffman code, and to extract the distribution of the data. Compute the Huffman code and compress the full dataset with it. Plot the results in comparison to other choices of input alphabets.

3.2 Part 2: LZ Coding

Viewing the dataset as a symbol stream (after re-formatting), use the training set to decide the best input alphabet of the LZ algorithm (you can choose between LZ77 and LZ78, but note your choice in the report). Compress the full dataset with the LZ algorithm. Plot the results in comparison to other choices of input alphabets.

3.3 Part 3: Structured Compression

Now examine the structure of the dataset and try to come up with a better compression than Parts 1,2. For example, if there are multiple fields, you may tailor the compression to each one separately. In this part you may also apply advanced techniques from the literature and/or formulate the compression problem as a machine-learning task (you can do anything with the data to find a good compressor, but remember to only use the training set). Plot the results and compare to Parts 1,2.

4 Deliverables

1. A final report describing the design+implementation, and showing the results. **Up to 5 pages.**
2. Your executable, source code, and instructions how to use them.
3. A mode of execution that accepts an external dataset file with the same structure of the datasets you used (to test your algorithm on similar datasets).

5 Need Help?

You are free to discuss ideas/questions with the course lecturer. You are also welcome to discuss between the teams.

6 Good Luck!