LargeScale Graph Compression using

Lempel Ziv Markov Algorithm

Arvind Murugan

Department of Computer Science

VSB – Technical University of Ostrava

# Introduction

In recent years, the volume of data handled across all domains have been increased rapidly and in future it will increase further more.

In graph theory, adjacency list or neighbors list means the list of connected nodes to that specific node.

Largescale networks like computer networks, social networks, mobile call networks, the WorldWideWeb network, protein regulation networks, contains a huge size of adjacency list for every node. And in future, it will increase rapidly. Representing the list and the data will become tedious and hard to run.

Using an Arithmetic compression technique like Delta Encoding and other encoding techniques can help us to overcome this problem, but only for larger files in sizes of MBs. But for storing files that are in sizes of TBs, they consume more time and cost.

One advantage of having larger ids is that the adjacency list will contain nodes whose ids gives us most matching patterns. Which is the key property of compression.

As a proposal to one of the solutions to storing the largescale graphs data in memory, I’ve used Lempel Ziv Markov Algorithm to compress the largescale graphs and provided the experimental result on the performance of the compression method.

To perform the analysis, I’ve used the python programming language in Jupyter Notebook. The function procedure is explained in the next section.

# Lempel Ziv Markov Algorithm

Compression is achieved by replacing repeated occurrences of data with references to a single copy of that data existing earlier in the uncompressed data stream.

## dESCRIPTION

* Compression is achieved by replacing repeated occurrences of data with references to a single copy of that data existing earlier in the uncompressed data stream.
* When a match is not found, the current input stream character is assumed to be the first character of an existing string in the dictionary (since the dictionary is initialized with all possible characters), so only the last matching index is output (which may be the pre-initialized dictionary index corresponding to the previous (or the initial) input character).
* A match is encoded by a pair of numbers called a length-distance pair, which is equivalent to the statement "each of the next length characters is equal to the characters exactly distance characters behind it in the uncompressed stream". (The distance is sometimes called the offset instead.)
* In LZMA compression, the compressed stream is a stream of bits, encoded using an adaptive binary range coder. The stream is divided into packets, each packet describing either a single byte, or an LZ77 sequence with its length and distance implicitly or explicitly encoded.
* Each part of each packet is modeled with independent contexts, so the probability predictions for each bit are correlated with the values of that bit (and related bits from the same field) in previous packets of the same type. Both the lzip[8] and the LZMA SDK documentation describes this stream format.

## pROCEDURE

* Put the edge text files in the “FILETOBECOMPRESSED” folder and run the application.
* Our application reads the file using pandas dataframe, then takes the edge list and generates adjacency list.
* For every node, the adjacency list dictionary, including both in and out vertices, is saved as LZMA compressed message after converting the dictionary from python dict to string.
* The adjacency list consisting of neighbors with closer id value, then it is more efficiently compressed.

# Dataset

Considering my system properties, I’ve chosen the below described three datasets from Stanford Large Network Dataset Collection - SNAP: Stanford (url: <https://snap.stanford.edu/data/index.html>).

The most suitable data for LZMA compression is the data that contains more matching sequences of symbols.

## Wiki Vote

The network contains all the Wikipedia voting data from the inception of Wikipedia till January 2008. Nodes in the network represent wikipedia users and a directed edge from node i to node j represents that user i voted on user j.

Wikipedia is a free encyclopedia written collaboratively by volunteers around the world. A small part of Wikipedia contributors are administrators, who are users with access to additional technical features that aid in maintenance.

In order for a user to become an administrator a Request for adminship (RfA) is issued and the Wikipedia community via a public discussion or a vote decides who to promote to adminship.

Using the latest complete dump of Wikipedia page edit history (from January 3 2008) we extracted all administrator elections and vote history data.

This gave us 2,794 elections with 103,663 total votes and 7,066 users participating in the elections (either casting a vote or being voted on).

Out of these 1,235 elections resulted in a successful promotion, while 1,559 elections did not result in the promotion. About half of the votes in the dataset are by existing admins, while the other half comes from ordinary Wikipedia users.

## Twitter:

This dataset consists of edges from all egonets combined from Twitter. Twitter data was crawled from public sources.

EgoNet (Egocentric Network Study Software) for the collection and analysis of egocentric social network data.

It helps the user to collect and analyse all the egocentric network data (all social network data of a website on the Internet), and provide general global network measures and data matrixes that can be used for further analysis by other software.

The egonet is the result of the links that it gives and receives certain address on the Internet and EgoNet is dedicated to collecting information about them and present it in a way useful to the users.

## Email eu Core network

This network represents the "core" of the email-EuAll network, which also contains links between members of the institution and people outside of the institution (although the node IDs are not the same).

This network was generated using email data from a large European research institution. They have anonymized information about all incoming and outgoing email between members of the research institution.

There is an edge (u, v) in the network if person u sent person v at least one email.

The e-mails only represent communication between institution members (the core), and the dataset does not contain incoming messages from or outgoing messages to the rest of the world.

# Experiment and Results

Every node contains the LZMA compressed message of their adjacency list including both the in vertices and out vertices.

LZMA help us to retrieve and update the list in the compressed form itself.

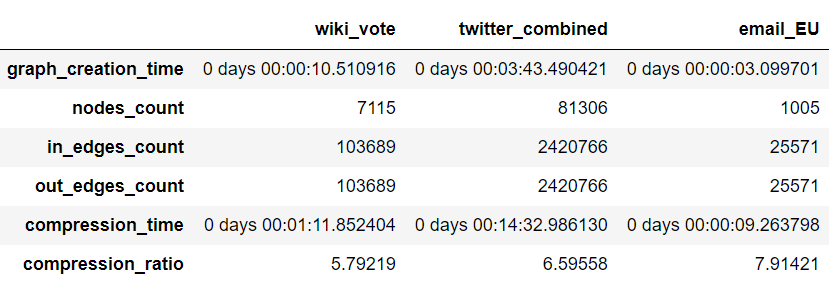


Figure 4‑1 LZMA Compression Report

# Conclusion

From the results, the compression time of LZMA is proportional to number of nodes and the edges in the graph. They play a vital role in time consumption.

LZMA also supports modifying the content of the compressed message at ease. This can be used to future development of this analysis.

Email EU network is the fastest compressed dataset compared to the others and has higher compression ratio. s

I finally conclude that it is clear that the Lempel Ziv Markov Algorithm compresses the largescale graph represented in adjacency list data structure with more compression rate.

# References

* Hao Yin, Austin R. Benson, Jure Leskovec, and David F. Gleich. "Local Higher-order Graph Clustering." In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017.
* J. Leskovec, J. Kleinberg and C. Faloutsos. Graph Evolution: Densification and Shrinking Diameters. ACM Transactions on Knowledge Discovery from Data (ACM TKDD), 1(1), 2007.
* J. McAuley and J. Leskovec. [Learning to Discover Social Circles in Ego Networks](http://i.stanford.edu/~julian/pdfs/nips2012.pdf). NIPS, 2012.
* J. Leskovec, D. Huttenlocher, J. Kleinberg. Signed Networks in Social Media. CHI 2010.
* J. Leskovec, D. Huttenlocher, J. Kleinberg. Predicting Positive and Negative Links in Online Social Networks. WWW 2010.