Efficient Data Storage and Machine Learning

Mirzakhmet Syzdykov[1]
[1]Satbayev University, Almaty, Kazakhstan
[1]mspmail598@gmail.com
[1]ORCID ID: https://orcid.org/0000-0002-8086-775X

Abstract. In this work we present to reader the novel research on account for efficiency of compression algorithms like Lempel-Ziv Welch and Aho-Corasick trees. We use them to build the proper storage which is called file system in a separate or generalized stream of data. These streams weren't adopted before for big data to be compressed and queried at a fast pace. We will show further that this is the most efficient model for storing arrays of data on a server end for a final file system. The efficient algorithm for Machine Learning on Aho-Corasick trees is also presented which performs the query in linear time without getting more time on the models like neural networks which are very hardware demanding nowadays. The data structure like trie by Turing Award winner Alfred V. Aho and Margaret J. Corasick remain of big potential in the present time and are subjected to extensive research in this work.

Keywords: trie, compression, storage, Machine Learning.

Introduction. The algorithm dated back to 1975 by Aho and Corasick was first proposed in [1]. The matching algorithm which is linear in performance was introduced in [2, 3] where the first case is about an Aho-Corasick data stream. The work by other authors [4] focuses mainly on memory performance within the multiple patterns matching which as we know can be done in parallel.

The simulation of Aho-Corasick machine is presented in [5]. This work is important to learn as it was first to use multiple pattern search – in this work the parallel algorithm is also presented where the search is adopted as a single thread.

Nathan et al. [6] give the application of the memory efficient algorithms for string searching on Aho-Corasick trees or simply tries.

The efficiency of the performance of Machine Learning (ML) queries is first discussed in [7] where it's given within the BigData.

MXNet [8] is another ML library based upon the neural network algorithms, the performance graphs are also given in the article.

The comparison of ML libraries is presented in [9] and as we can see the efficiency still doesn't converge to the final value as the most of the programs are experimental.

Summarizing all the above we can conclude that efficient memory storages like file systems and Automated Machine Learning (AML) become more demanding nowadays, thus, we have to implement more efficient algorithms for Machine Learning Querying (MLQ) and Compressed File System (CFS) in which the Lempel-Ziv Welch (LZW) automaton is realized within the Aho-Corasick tree method of compensation of the repeated occurrences of the pattern.

We will also state the main theorems regarding the MLQ and CFS which simplifies the definition of the final complexity of the usage of these engines as software packages.

As we have stated before the repeated occurrence of pattern, which can be dualized as a single or multiple incoming query to system, should be of the minimal possible complexity from O(1) to O(n) where O(n) stands for the complexity per the query of n-words, thus giving same O(1) complexity per each of them. This result presented in this work gives the possibility of the assumption of the upper bound of performance for MLQ and CFS or, at least, such definitions are to be definitely made – we will show it and give the proofs in our next sections.

The assumption of CFS as a modified Aho-Corasick tree is a classical approach of giving its definition to broad audience which consist of not only researchers, but developers also as Application Programming Interface (API) for these engines becomes more available due to the free license.

References

1. Aho, Alfred V., and Margaret J. Corasick. "Efficient string matching: an aid to bibliographic search." *Communications of the ACM* 18.6 (1975): 333-340.
2. Tao, Tao, and Amar Mukherjee. "Multiple-Pattern Matching In LZW Compressed Files Using Aho-Corasick Algorithm." *DCC*. 2005.
3. Aldwairi, Monther, Abdulmughni Y. Hamzah, and Moath Jarrah. "MultiPLZW: A novel multiple pattern matching search in LZW-compressed data." *Computer Communications* 145 (2019): 126-136.
4. Liangxu, Sun, and Li Linlin. "Improve Aho-Corasick algorithm for multiple patterns matching memory efficiency optimization." *J. Convergence Inf. Technol.(JCIT)* 7 (2012): 19.
5. Kida, Takuya, et al. "Multiple pattern matching in LZW compressed text." *Proceedings DCC'98 Data Compression Conference (Cat. No. 98TB100225). IEEE*, 1998.
6. Tuck, Nathan, et al. "Deterministic memory-efficient string matching algorithms for intrusion detection." *IEEE INFOCOM 2004*. Vol. 4. IEEE, 2004.
7. Al-Jarrah, Omar Y., et al. "Efficient machine learning for big data: A review." *Big Data Research* 2.3 (2015): 87-93.
8. Chen, Tianqi, et al. "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems." *arXiv preprint arXiv*:1512.01274 (2015).
9. Nguyen, Hoang, et al. "Efficient machine learning models for prediction of concrete strengths." *Construction and Building Materials* 266 (2021): 120950.