The Journal of Open Source Software

OrpailleCC: a Library for Data Stream Analysis on Embedded Systems

Martin Khannouz[1], **Bo Li**[1], **and Tristan Glatard**[1]

1 Department of Computer Science and Software Engineering, Concordia University, Montreal, Canada

DOI: 10.21105/joss.01485

Software

- Review
- Repository
- Archive

Submitted: 29 March 2019
Published: 25 July 2019

Summary

The Internet of Things could benefit in several ways from mining data streams on connected objects rather than in the cloud. In particular, limiting network communication with cloud services would improve user privacy and reduce energy consumption in connected devices. Besides, applications could leverage the computing power of connected objects for improved scalability.

OrpailleCC provides a consistent collection of data stream algorithms developed to be deployed on embedded devices. Its main objective is to support research on data stream mining for connected objects, by facilitating the comparison and benchmarking of algorithms in a consistent framework. It also enables programmers of embedded systems to use out-of-the-box algorithms with an efficient implementation. To the best of our knowledge, existing libraries of stream mining algorithms cannot be used on connected objects due to their resource consumption or assumptions about the target system (e.g., existence of a `malloc` function). Nevertheless, for more powerful devices such as desktop computers, Java frameworks such as Massive Online Analysis (Bifet, Holmes, Kirkby, & Pfahringer, 2010) and WEKA (Hall et al., 2009) achieve similar goals as OrpailleCC.

OrpailleCC targets the classes of problems discussed by Kejariwal, Kulkarni, & Ramasamy (2015), in particular Sampling and Filtering. Sampling covers algorithms that build a representative sample of a data stream. OrpailleCC implements the reservoir sampling (Vitter, 1985) and one variant, the chained reservoir sampling (Babcock, Datar, & Motwani, 2002). Filtering algorithms remove the stream elements that do not belong to a specific set. OrpailleCC implements the Bloom Filter (Bloom, 1970) and the Cuckoo Filter (Fan, Andersen, Kaminsky, & Mitzenmacher, 2014), two well-tested algorithms that address this problem.

In addition to Sampling and Filtering, OrpailleCC provides algorithms for stream Classification and for stream Compression. The Micro-Cluster Nearest Neighbour algorithm (Tennant, Stahl, Rana, & Gomes, 2017) is based on the k-nearest neighbor to classify a data stream while detecting concept drifts. The Lightweight Temporal Compression (Schoellhammer, Greenstein, Osterweil, Wimbrow, & Estrin, 2004) and a multi-dimensional variant (Li, Sarbishei, Nourani, & Glatard, 2018) are two methods to compress data streams.

All implementations rely as little as possible on functions provided by the operating system, for instance `malloc`, since such functions are typically not available on embedded systems. When algorithms cannot be implemented without such functions, the library uses template parameters to request the required functions from the user. All algorithms are developed for FreeRTOS (Amazon Web Services, n.d.), a free real-time operating system used in embedded systems, but they should work on any micro-controller with a C++11 compiler. The C++11 programming language was chosen for its performance as well as its popularity in the field. All methods are tested and tests are run through Travis-CI.

In the future, we plan to extend the library with other reliable algorithms to widely cover as many common problems as possible. We also plan to use it as a basis to design new stream classification methods. External contributions are, of course, most welcome.

References

Amazon Web Services. (n.d.). FreeRTOS. https://www.freertos.org/.

Babcock, B., Datar, M., & Motwani, R. (2002). Sampling from a moving window over streaming data. In *Proceedings of the thirteenth annual Association for Computing Machinery-SIAM Symposium on Discrete algorithms* (pp. 633–634). Society for Industrial; Applied Mathematics.

Bifet, A., Holmes, G., Kirkby, R., & Pfahringer, B. (2010). MOA: Massive Online Analysis. *Journal of Machine Learning Research*, *11*(May), 1601–1604.

Bloom, B. H. (1970). Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, *13*(7), 422–426. doi:10.1145/362686.362692

Fan, B., Andersen, D. G., Kaminsky, M., & Mitzenmacher, M. D. (2014). Cuckoo filter. In *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies - CoNEXT 14*. ACM Press. doi:10.1145/2674005.2674994

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software. *ACM SIGKDD Explorations Newsletter*, *11*(1), 10. doi:10.1145/1656274.1656278

Kejariwal, A., Kulkarni, S., & Ramasamy, K. (2015). Real time analytics. *Proceedings of the VLDB Endowment*, *8*(12), 2040–2041. doi:10.14778/2824032.2824132

Li, B., Sarbishei, O., Nourani, H., & Glatard, T. (2018). A multi-dimensional extension of the Lightweight Temporal Compression method. In *2018 IEEE International Conference on Big Data (big data)*. IEEE. doi:10.1109/bigdata.2018.8621946

Schoellhammer, T., Greenstein, B., Osterweil, E., Wimbrow, M., & Estrin, D. (2004). Lightweight Temporal Compression of Microclimate Datasets [wireless sensor networks]. In *29th annual IEEE International Conference on Local Computer Networks*. IEEE (Comput. Soc.). doi:10.1109/lcn.2004.72

Tennant, M., Stahl, F., Rana, O., & Gomes, J. B. (2017). Scalable real-time classification of data streams with concept drift. *Future Generation Computer Systems*, *75*, 187–199. doi:10.1016/j.future.2017.03.026

Vitter, J. S. (1985). Random sampling with a reservoir. *ACM Transactions on Mathematical Software*, *11*(1), 37–57. doi:10.1145/3147.3165