The Journal of Open Source Software

mlpack 3: a fast, flexible machine learning library

Ryan R. Curtin[1], **Marcus Edel**[2], **Mikhail Lozhnikov**[3], **Yannis Mentekidis**[1], **Sumedh Ghaisas**[1], **and Shangtong Zhang**[4]

1 Center for Advanced Machine Learning, Symantec Corporation **2** Institute of Computer Science, Free University of Berlin **3** Moscow State University, Faculty of Mechanics and Mathematics **4** University of Alberta

DOI: 10.21105/joss.00726

Software

- Review
- Repository
- Archive

Submitted: 24 April 2018
Published: 07 May 2018

Summary

In the past several years, the field of machine learning has seen an explosion of interest and excitement, with hundreds or thousands of algorithms developed for different tasks every year. But a primary problem faced by the field is the ability to scale to larger and larger data—since it is known that training on larger datasets typically produces better results (Halevy, Norvig, and Pereira 2009). Therefore, the development of new algorithms for the continued growth of the field depends largely on the existence of good tooling and libraries that enable researchers and practitioners to quickly prototype and develop solutions (Sonnenburg et al. 2007). Simultaneously, useful libraries must also be efficient and well-implemented. This has motivated our development of mlpack.

mlpack is a flexible and fast machine learning library written in C++ that has bindings that allow use from the command-line and from Python, with support for other languages in active development. mlpack has been developed actively for over 10 years (Curtin et al. 2011, Curtin, Cline, et al. (2013)), with over 100 contributors from around the world, and is a frequent mentoring organization in the Google Summer of Code program (https://summerofcode.withgoogle.com). If used in C++, the library allows flexibility with no speed penalty through policy-based design and template metaprogramming (Alexandrescu 2001); but bindings are available to other languages, which allow easy use of the fast mlpack codebase.

For fast linear algebra, mlpack is built on the Armadillo C++ matrix library (Sanderson and Curtin 2016), which in turn can use an optimized BLAS implementation such as OpenBLAS (Xianyi, Qian, and Saar 2018) or even NVBLAS (NVIDIA 2015) which would allow mlpack algorithms to be run on the GPU. In order to provide fast code, template metaprogramming is used throughout the library to reduce runtime overhead by performing any possible computations and optimizations at compile time. An automatic benchmarking system is developed and used to test the efficiency of mlpack's algorithms (Edel, Soni, and Curtin 2014).

mlpack contains a number of standard machine learning algorithms, such as logistic regression, random forests, and k-means clustering, and also contains cutting-edge techniques such as a compile-time optimized deep learning and reinforcement learning framework, dual-tree algorithms for nearest neighbor search and other tasks (Curtin, March, et al. 2013), a generic optimization framework with numerous optimizers (Curtin et al. 2017), a generic hyper-parameter tuner, and other recently published machine learning algorithms.

For a more comprehensive introduction to mlpack, see the website at http://www.mlpack.org/ or a recent paper detailing the design and structure of mlpack (Curtin and Edel 2017).

References

Alexandrescu, A. 2001. *Modern C++ Design: Generic Programming and Design Patterns Applied*. Addison-Wesley.

Curtin, R.R., and M. Edel. 2017. "Designing and Building the Mlpack Open-Source Machine Learning Library." *arXiv Preprint arXiv:1708.05279*.

Curtin, R.R., S. Bhardwaj, M. Edel, and Y. Mentekidis. 2017. "A Generic and Fast C++ Optimization Framework." *arXiv Preprint arXiv:1711.06581*.

Curtin, R.R., J.R. Cline, N.P. Slagle, M.L. Amidon, and A.G. Gray. 2011. "mlpack: A Scalable C++ Machine Learning Library." In *BigLearning: Algorithms, Systems, and Tools for Learning at Scale*.

Curtin, R.R., J.R. Cline, N.P. Slagle, W.B. March, P. Ram, N.A. Mehta, and A.G. Gray. 2013. "mlpack: A Scalable C++ Machine Learning Library." *Journal of Machine Learning Research* 14:801–5.

Curtin, R.R., W.B. March, P. Ram, D.V. Anderson, A.G. Gray, and C.L. Isbell Jr. 2013. "Tree-Independent Dual-Tree Algorithms." In *Proceedings of the 30th International Conference on Machine Learning (Icml '13)*, 1435–43.

Edel, M., A. Soni, and R.R. Curtin. 2014. "An Automatic Benchmarking System." In *Proceedings of the Nips 2014 Workshop on Software Engineering for Machine Learning*.

Halevy, A., P. Norvig, and F. Pereira. 2009. "The Unreasonable Effectiveness of Data." *IEEE Intelligent Systems* 24 (2). IEEE:8–12.

NVIDIA. 2015. "NVBLAS Library.";http://docs.nvidia.com/cuda/nvblas.

Sanderson, C., and R.R. Curtin. 2016. "Armadillo: A Template-Based C++ Library for Linear Algebra." *Journal of Open Source Software*. Journal of Open Source Software.

Sonnenburg, S., M.L. Braun, C.S. Ong, S. Bengio, L. Bottou, G. Holmes, Y. LeCun, et al. 2007. "The Need for Open Source Software in Machine Learning." *Journal of Machine Learning Research* 8 (Oct):2443–66.

Xianyi, Z., W. Qian, and W. Saar. 2018. "OpenBLAS: An Optimized BLAS Library." http://www.openblas.net.