

Original software publication

LIBS2ML: A library for scalable second order machine learning algorithms

Vinod Kumar Chauhan^{a,b,*}, Anuj Sharma^b, Kalpana Dahiya^c

^a Institute for Manufacturing, Department of Engineering, University of Cambridge, Alan Reece Building, 17 Charles Babbage Rd, Cambridge CB3 0FS, UK

^b Department of Computer Science and Applications, Panjab University Chandigarh, India

^c University Institute of Engineering and Technology, Panjab University Chandigarh, India

ARTICLE INFO

Keywords:

Stochastic optimization
Second order methods
Large-scale machine learning

ABSTRACT

Most of the machine learning libraries are either in MATLAB/Python/R which are very slow and not suitable for large-scale learning, or are in C/C++ which does not have easy ways to take input and display results. LIBS2ML¹ has been developed using MEX files, i.e., C++ with MATLAB/Octave interface to take the advantage of faster learning using C++ and easy I/O using MATLAB/Octave. So, LIBS2ML is a completely unique due to its focus on the scalable second order methods – the hot research topic – and being based on MEX files. It provides researchers a comprehensive environment to evaluate their ideas and it also provides machine learning practitioners an effective tool to deal with the large-scale learning problems. LIBS2ML is an open-source, highly efficient, extensible, scalable, readable, portable and easy to use library.

Code metadata

Current code version	1.0
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2021-97
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/6078990/tree/v1
Legal Code License	Apache 2.0 open source license
Code versioning system used	None
Software code languages, tools, and services used	MEX files (MATLAB or Octave and C++ compiler)
Compilation requirements, operating environments & dependencies	Any operating system with MATLAB or Octave and compatible C++ compiler
If available Link to developer documentation/manual	https://github.com/jmdvinodjmd/LIBS2ML/blob/master/README.md
Support email for questions	vk359@cam.ac.uk

1. Introduction

In this library, we have solved the commonly used empirical risk minimization (ERM) problem in machine learning, as given below:

$$\min_w F(w) = \frac{1}{n} \sum_{i=1}^n f(w; x_i, y_i) = \frac{1}{n} \sum_{i=1}^n f_i(w), \quad (1)$$

where $w \in \mathbb{R}^d$ is a parameter vector, d is the number of features, $f_i(w) : \mathbb{R}^d \rightarrow \mathbb{R}$ is a loss function and $\{(x_i, y_i)\}_{i=1}^n$ is the training set with n data points. ERM is a general class of problems and by

substituting different loss functions, we get different problems like, SVM, logistic regression and ridge regression etc.

Nowadays big data is one of the major challenge in machine learning [1,2] and stochastic optimization techniques have been quite effective to tackle the challenge [3]. After the success of stochastic first order methods [4,5], the focus of learning algorithms have shifted towards the stochastic second order methods due to their faster convergence rates and availability of computing resources to deal with their high computational complexities [6].

The code (and data) in this article has been certified as Reproducible by Code Ocean: (<https://codeocean.com/>). More information on the Reproducibility Badge Initiative is available at <https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals>.

* Corresponding author at: Institute for Manufacturing, Department of Engineering, University of Cambridge, Alan Reece Building, 17 Charles Babbage Rd, Cambridge CB3 0FS, UK.

E-mail addresses: vk359@cam.ac.uk (V.K. Chauhan), anuj@pu.ac.in (A. Sharma), kalpanas@pu.ac.in (K. Dahiya).

URLs: <https://sites.google.com/site/jmdvinodjmd/> (V.K. Chauhan), <https://anuj-sharma.in/> (A. Sharma).

¹ LIBS2ML is continuously extended by adding more problems and methods.

<https://doi.org/10.1016/j.simpa.2021.100123>

Received 13 August 2021; Received in revised form 21 August 2021; Accepted 22 August 2021

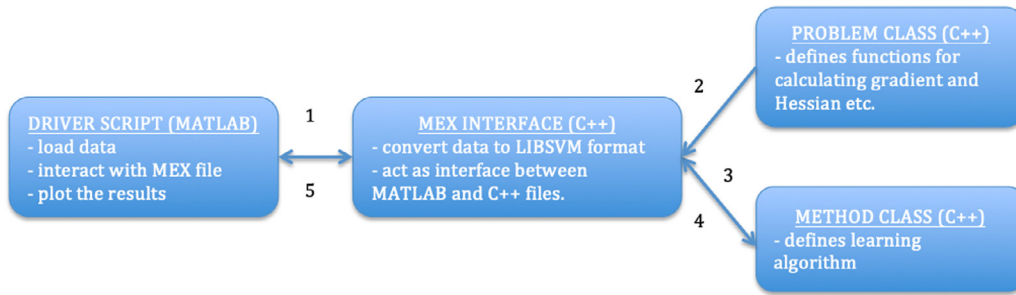


Fig. 1. Architecture consists of driver script, interface, problem and method classes.

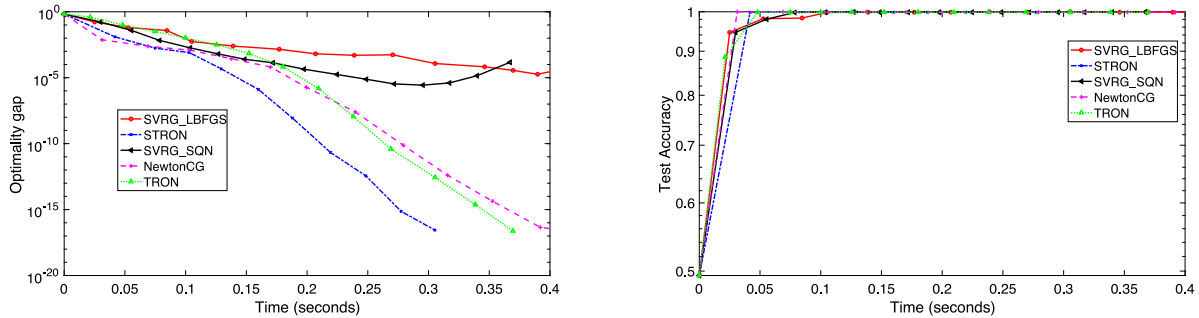


Fig. 2. Left part presents optimality versus training time (in seconds) and right part presents accuracy versus training time, on mushroom dataset using logistic regression.

LIBS2ML has been developed due to great scope of scalable second order methods and to make the task of researchers and practitioners easy. MATLAB/Python/R based libraries are easy to use, easy to code and it is easy to explore new ideas but these are very slow as compared with low level languages and so these are not suitable to deal with large-scale problems, e.g. SGDLibrary [7]. On the other hand, C/C++ based libraries are fast and good to solve large-scale problems but they are difficult to code and lack functionality to show convergence of algorithms, e.g. LIBLINEAR [8]. LIBS2ML combines the best of these two programming worlds, where all learning algorithms are implemented using C++ and MATLAB/Octave is used to take input and display results.

2. LIBS2ML

LIBS2ML has been designed using a modular approach for flexibility and extensibility. The library separates problem, method, I/O and auxiliary code into different classes and folders. Each problem like, logistic regression, is contained in a separate C++ file which defines gradients, Hessian and related calculations. Similarly, each learning method is defined in a separate C++ file. Fig. 1 depicts the high level architecture of LIBS2ML, which shows four components: First is MATLAB/Octave driver script which performs I/O, second is a MEX (MATLAB executable) file which acts as an interface between MATLAB and C++, third and fourth components are problem and method classes, which are developed in C++. Typical execution flow is denoted in the figure with numbers where first driver script loads data and other information, and passes to the MEX file which act as interface between different components. Next, MEX file loads the problem instance and pass problem, data and other information to an instance of method. The method performs the training and returns results to the MEX file which then passes to driver script. The driver script displays the results.

The design of the library provides following features:

Highly Efficient: All the learning algorithms are developed in C++.

Modular Design: The library separates problems, methods and I/O operations.

Extensibility: New problem or method can be added by adding single new file.

Portability: The library compiles and runs on MATLAB/Octave so it is portable and can work on any platform (Windows, Mac OS and Linux etc.).

Easy: The library is easy to install, use and understand, and has no dependency on other libraries.

LIBS2ML uses MATLAB data formats for the convenience of users. The datasets can be converted to MATLAB format using LIBSVM library [9].

3. Empirical results and practical usage

MATLAB/Octave with a compatible C/C++ compiler are the pre-requisites for LIBS2ML. Moreover, we need to develop a driver script in MATLAB/Octave to solve a problem using the library, where we load the desired dataset, call the MEX interface with information about the problem to be solved, learning method, data and other required parameters, and plot the results in the desired format. Example scripts along with README file containing documentation are provided in the library to use all its functionality.

Empirical results for solving l_2 -regularized logistic regression problem using TRON [10], Newton-CG [11], STRON [6], SVRG-SQN [12] and SVRG-LBFGS [13] methods with mushroom dataset are provided in Fig. 2. The results are plotted as optimality and test accuracy against the training time (in seconds) of learning algorithms. The results can be plotted against number of epochs but that is not important for large-scale learning problems because small and large-scale learning have different trade-offs, and time is a major factor for large-scale learning [14].

4. Impact

LIBS2ML is a unique library due to its focus on scalable second order methods – the hot research topic [2,6] – for solving large-scale learning problems, and utilizing best of MATLAB/Octave and C++. It is a modular, easily understandable, extensible, portable, highly efficient and specially designed to deal with the big data challenge in machine

learning. It is helpful for beginners as well as advanced researchers to explore new ideas, and to the machine learning practitioners.

LIBS2ML was used to study scalable second order methods to solve large-scale machine learning problems in [6]. [6] utilizes best of stochastic and full-batch regimes to solve large-scale problems using inexact Newton method.

Acknowledgments

Most of this work was done at Panjab University Chandigarh when first author was pursuing his PhD. First author is also thankful to Ministry of Human Resource Development, Government of INDIA, to provide fellowship (University Grants Commission—Senior Research Fellowship) to pursue his Ph.D. work.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] Vinod Kumar Chauhan, Kalpana Dahiya, Anuj Sharma, Problem formulations and solvers in linear svm: A review, *Artif. Intell. Rev.* (2018).
- [2] Vinod Kumar Chauhan, *Stochastic Optimization for Large-Scale Machine Learning*, first ed., CRC Press, ISBN: 9781032131757, 2021.
- [3] Vinod Kumar Chauhan, Anuj Sharma, Kalpana Dahiya, Saags: Biased stochastic variance reduction methods for large-scale learning, *Appl. Intell.* 49 (9) (2019) 3331–3361.
- [4] Vinod Kumar Chauhan, Kalpana Dahiya, Anuj Sharma, Mini-batch block-coordinate based stochastic average adjusted gradient methods to solve big data problems, in: *Proceedings of the Ninth Asian Conference on Machine Learning*, Vol. 77, PMLR, 2017, pp. 49–64.
- [5] Vinod Kumar Chauhan, Anuj Sharma, Kalpana Dahiya, Faster learning by reduction of data access time, *Appl. Intell.* 48 (12) (2018) 4715–4729.
- [6] Vinod Kumar Chauhan, Anuj Sharma, Kalpana Dahiya, Stochastic trust region inexact Newton method for large-scale machine learning, *Int. J. Mach. Learn. & Cyber.* 11 (7) (2020) 1541–1555.
- [7] Hiroyuki Kasai, Sgdlibrary: A matlab library for stochastic optimization algorithms, *J. Mach. Learn. Res.* 18 (1) (2017) 7942–7946.
- [8] R. Fan, K. Chang, C. Hsieh, X. Wang, C. Lin, Liblinear: A library for large linear classification, *JMLR* 9 (2008) 1871–1874.
- [9] Chih-Chung Chang, Chih-Jen Lin, LIBSVM: A library for support vector machines, *ACM Trans. Intell. Syst. Technol. (TIST)* 2 (2011) 27, 27:1–27.
- [10] Chih-Yang Hsia, Wei-Lin Chiang, Chih-Jen Lin, Preconditioned conjugate gradient methods in truncated newton frameworks for large-scale linear classification, in: *Proceedings of the Tenth Asian Conference on Machine Learning*, *Proceedings of Machine Learning Research*, PMLR, 2018.
- [11] R. Byrd, G. Chin, W. Neveitt, J. Nocedal, On the use of stochastic hessian information in optimization methods for machine learning, *SIAM J. Optim.* 21 (3) (2011) 977–995.
- [12] Philipp Moritz, Robert Nishihara, Michael I. Jordan, A linearly-convergent stochastic l-bfgs algorithm, in: *AISTATS*, 2016.
- [13] Ritesh Kolte, Murat Erdogdu, Ayfer Ozgur, Accelerating svrg via second-order information, in: *NIPS Workshop on Optimization for Machine Learning*, 2015.
- [14] Léon Bottou, Olivier Bousquet, The tradeoffs of large-scale learning, *Optim. Mach. Learn.* (2011) 351.