

Singapore Management University Institutional Knowledge at Singapore Management University

Research Collection School Of Information Systems

School of Information Systems

10-2017

SOL: A library for scalable online learning algorithms

Yue WU

University of Science and Technology of China

Steven C. H. HOI

Singapore Management University, CHHOI@smu.edu.sg

Chenghao LIU

Singapore Management University, chliu@smu.edu.sg

Jing LU

Singapore Management University, jing.lu.2014@phdis.smu.edu.sg


Doyen SAHOO

Singapore Management University, doyens@smu.edu.sg

See next page for additional authors

DOI: <https://doi.org/10.1016/j.neucom.2017.03.077>

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research

 Part of the [Online and Distance Education Commons](#), and the [Theory and Algorithms Commons](#)

Citation

WU, Yue; HOI, Steven C. H.; LIU, Chenghao; LU, Jing; SAHOO, Doyen; and YU, Nenghai. SOL: A library for scalable online learning algorithms. (2017). *Neurocomputing*. 260, 9-12. Research Collection School Of Information Systems.

Available at: https://ink.library.smu.edu.sg/sis_research/3991

This Journal Article is brought to you for free and open access by the School of Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email libIR@smu.edu.sg.

Author

Yue WU, Steven C. H. HOI, Chenghao LIU, Jing LU, Doyen SAHOO, and Nenghai YU

SOL: A library for scalable online learning algorithms

Yue Wu^{a,b}, Steven C.H. Hoi^{b,*}, Chenghao Liu^{b,c}, Jing Lu^b, Doyen Sahoo^b, Nenghai Yu^a

^aSchool of Information Science and Technology, University of Science and Technology of China, China

^bSchool of Information Systems, Singapore Management University, Singapore

^cSchool of Computer Science and Technology, Zhejiang University, China

ARTICLE INFO

Article history:

Received 16 September 2016

Accepted 31 March 2017

Available online 15 April 2017

Communicated by Prof. Zidong Wang

Keywords:

Online learning

Scalable machine learning

High dimensionality

Sparse learning

ABSTRACT

SOL is an open-source library for scalable online learning with high-dimensional data. The library provides a family of regular and sparse online learning algorithms for large-scale classification tasks with high efficiency, scalability, portability, and extensibility. We provide easy-to-use command-line tools, python wrappers and library calls for users and developers, and comprehensive documents for both beginners and advanced users. SOL is not only a machine learning toolbox, but also a comprehensive experimental platform for online learning research. Experiments demonstrate that SOL is highly efficient and scalable for large-scale learning with high-dimensional data.

Software metadata

(executable) Software metadata description

Current software version	v1.1.0
Permanent link to executables of this version	https://github.com/Neurocomputing/NEUCOM-D-16-02987
Legal Software License	Apache 2.0 open source license
Computing platform / Operating system	Linux, OS X, Windows.
Installation requirements & dependencies	C++11 Compiler, (Python 2.7)
Link to user manual	https://github.com/libol/sol/wiki
Support email for questions	chhoi@smu.edu.sg

Code metadata

Code metadata description

Current code version	v1.1.0
Permanent link to code/repository used of this code version	https://github.com/Neurocomputing/NEUCOM-D-16-02987
Legal Code License	Apache 2.0 open source license
Code versioning system used	git
Software code languages, tools, and services used	C++/Python
Compilation requirements, operating environments & dependencies	GCC/MSVC/Clang, Python2.7
If available Link to developer documentation/manual	https://github.com/libol/sol/wiki
Support email for questions	chhoi@smu.edu.sg

* Corresponding author at: Steven C.H. Hoi, School of Information Systems, Singapore Management University, 80 Stamford Road, Singapore 178902.

E-mail addresses: wye@mail.ustc.edu.cn (Y. Wu), chhoi@smu.edu.sg (S.C.H. Hoi), twinsken@zju.edu.cn (C. Liu), jing.lu.2014@phdis.smu.edu.sg (J. Lu), doyen.2014@phdis.smu.edu.sg (D. Sahoo), ynh@ustc.edu.cn (N. Yu).

1. Introduction

In the era of big data, data is large not only in sample size, but also in feature/dimension size, e.g., web-scale text classification with millions of dimensions. Traditional batch learning algorithms fall short in low efficiency and poor scalability, e.g., high memory consumption and expensive cost for re-training new data. Online learning represents a family of efficient and scalable algorithms that sequentially learn one example at a time. Some existing toolbox, e.g., LIBOL [1], allows researchers in academia to benchmark different online learning algorithms, but it was not designed for practical developers to tackle online learning with large-scale high-dimensional data in industry.

In this work, we develop SOL as an easy-to-use scalable online learning toolbox for large-scale binary and multi-class classification tasks. It includes a family of ordinary and sparse online learning algorithms, and is highly efficient and scalable for processing high-dimensional data by using (i) parallel threads for both loading and learning the data, and (ii) specially designed data structure for high-dimensional data. The library is implemented in standard C++ with the cross platform ability and there is no dependency on other libraries. To facilitate developing new algorithms, the library is carefully designed and documented with high extensibility. We also provide python wrappers to facilitate experiments and library calls for advanced users. The SOL website is host at <http://sol.stevenhoi.org> and the software is made available <https://github.com/libol/sol>.

2. Scalable online learning for large-scale linear classification

2.1. Overview

Online learning operates sequentially to process one example at a time. Consider $\{(\mathbf{x}_t, y_t) | t \in [1, T]\}$ be a sequence of training data examples, where $\mathbf{x}_t \in \mathbb{R}^d$ is a d -dimensional vector, $y_t \in \{+1, -1\}$ for binary classification or $y_t \in \{0, \dots, C-1\}$ for multi-class classification (C classes). As Algorithm 1 shows, at each time step t , the learner receives an incoming example \mathbf{x}_t and then predicts the scores \hat{y}_t over classes. Afterward, the true label y_t is revealed and the learner suffers a loss $\ell(y_t, \hat{y}_t)$, e.g., the hinge loss is commonly used $\ell(y_t, \hat{y}_t) = \max(0, 1 - y_t \cdot \hat{y}_t)$ for binary classification. For sparse online learning, one can modify the loss with $L1$ regularization $\ell(y_t, \hat{y}_t) + \lambda \|\mathbf{w}_t\|_1$ to induce sparsity for the learned model \mathbf{w} . At the end of each learning step, the learner decides when and how to update the model.

Algorithm 1: SOL: Online Learning Framework for Linear Classification.

```

Initialize:  $\mathbf{w}_1 = \mathbf{0}$ ;
for  $t = 1; t \leq T; ++t$  do
  Receive  $\mathbf{x}_t \in \mathbb{R}^d$ , predict  $\hat{y}_t$ , receive true label  $y_t$ ;
  Suffer loss  $\ell(y_t, \hat{y}_t)$ ;
  if  $\ell(y_t, \hat{y}_t)$  then
    |  $\mathbf{w}_{t+1} \leftarrow \text{update}(\mathbf{w}_t)$ ;
  end
end

```

The goal of our work is to implement most state-of-the-art online learning algorithms to facilitate research and application purposes on the real world large-scale high dimensional data. Especially, we include sparse online learning algorithms which can effectively learn important features from the high dimensional real world data [2]. We provide algorithms for both binary and multi-class problems. These algorithms can also be classified into first order algorithms [3] and second order algorithms [4] from

Table 1

Summary of the implemented online learning algorithms in SOL.

Type	Methodology	Algorithm	Description
Online learning	First order	Perceptron [5]	The perceptron algorithm
		OGD [6]	Online gradient descent
		PA [7]	Passive aggressive algorithms
	Second order	ALMA [8]	Approximate large margin algorithm
		RDA [3]	Regularized dual averaging
		SOP [9]	Second-order perceptron
		CW [10]	Confidence weighted learning
		ECCW [11]	Exactly convex confidence weighted learning
		AROW [4]	Adaptive regularized online learning
		Ada-FOBOS [12]	Adaptive gradient descent
Ada-RDA [12]	Adaptive regularized dual averaging		
Sparse online learning	First order	STG [2]	Sparse online learning via truncated gradient
		FOBOS-L1 [13]	l_1 regularized forward backward splitting
		RDA-L1 [3]	Mixed l_1/l_2 regularized dual averaging
	Second order	ERDA-L1 [3]	Enhanced l_1/l_2 regularized dual averaging
		Ada-FOBOS-L1 [12]	Ada-FOBOS with l_1 regularization
		Ada-RDA-L1 [12]	Ada-RDA with l_1 regularization

the model’s perspective. The implemented algorithms are listed in Table 1.

2.2. The software package

The SOL package includes a library, command-line tools, and python wrappers for the learning task. SOL is implemented in standard C++ to be easily compiled and built in multiple platforms (Linux, Windows, MacOS, etc.) without dependency. It supports “libsvm” and “csv” data formats. To accelerate the training process, a binary format is defined. SOL is released under the Apache 2.0 open source license.

2.2.1. Practical usage

To illustrate the training and testing procedure, we use the OGD algorithm with a constant learning rate 1 to learn a model for the “rcv1”¹ dataset and save the model to “rcv1.model”.

```

$ sol_train --params eta=1 -a ogd rcv1_train rcv1.model
[output skipped]
$ sol_test rcv1.model rcv1_test predict.txt
test accuracy: 0.9545

```

We can also use the python wrappers to train the same model. The wrappers provide the cross validation ability which can be used to select the best parameters as the following commands show. More advanced usages of SOL can be found in the documentation.

```

$ sol_train.py --cv eta=0.25:2:128 -a ogd rcv1_train rcv1.model
cross validation parameters: [(‘eta’, 32.0)]
$ sol_test.py rcv1.model rcv1_test predict.txt
test accuracy: 0.9744

```

2.2.2. Documentation and design

The SOL package comes with detailed documentation. The README file gives an “Installation” section for different platforms,

¹ <https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#rcv1.binary>

Table 2
Comparison of SOL with LIBLINEAR and VW on “rcv1”.

Algorithm	Train time(s)	Accuracy	Algorithm	Train time(s)	Accuracy
Perceptron	8.4296 ± 0.0867	0.9625 ± 0.0014	OGD	8.4109 ± 0.0982	0.9727 ± 0.0006
PA	8.4506 ± 0.1031	0.9649 ± 0.0015	PA1	8.5113 ± 0.1143	0.9760 ± 0.0005
PA2	8.4445 ± 0.1068	0.9758 ± 0.0003	ALMA	9.1464 ± 0.1624	0.9745 ± 0.0009
RDA	8.4809 ± 0.0899	0.9212 ± 0.0000	ERDA	8.4623 ± 0.1123	0.9493 ± 0.0002
CW	8.4356 ± 0.1118	0.9656 ± 0.0010	ECCW	8.4641 ± 0.1116	0.9681 ± 0.0009
SOP	8.5246 ± 0.1017	0.9627 ± 0.0012	AROW	8.4390 ± 0.1292	0.9766 ± 0.0002
Ada-FOBOS	8.4897 ± 0.0872	0.9769 ± 0.0003	Ada-RDA	8.4388 ± 0.1140	0.9767 ± 0.0003
VW	11.3581 ± 0.3423	0.9754 ± 0.0009	LIBLINEAR	77.9274 ± 1.4742	0.9771 ± 0.0000

```

Vector<float> w; //weight vector
void Iterate(SVector<float> x, int y) {
    //predict label with dot product
    float predict = dotmul(w, x);
    float loss = max(0, 1 - y * predict); //hinge loss
    if (loss > 0) { //non-zero loss, update the model
        w = w + eta * y * x; //eta is the learning rate
        //calculate the L2 norm of weight vector
        float w_norm = Norm2(w);
        if (w_norm > 1) w /= w_norm;
    }
}

```

Fig. 1. Example code to implement the core function of “ALMA” algorithm.

and a “Quick Start” section as a basic tutorial to use the package for training and testing. We also provide online Wiki for advanced users. Users who want to have a comprehensive evaluation of online algorithms and parameter settings can refer to the “Command” section. If users want to call the library in their own project, they can refer to the “Library” section. For those who want to implement a new algorithm, they can read the “Design” section and the “Extension Examples” section. The whole package is designed for high efficiency, scalability, portability, and extensibility.

- **Efficiency:** it is implemented in C++ and optimized to reduce time and memory cost.
- **Scalability:** Data samples are stored in a sparse structure. All operations are optimized around the sparse data structure.
- **Portability:** All the codes follow the C++11 standard, and there is no dependency on external libraries. We use “cmake” to organize the project so that users on different platforms can build the library easily. SOL thus can run on almost every platform.
- **Extensibility:** (i) the library is written in a modular way, including *PARIO*(for PARAllel IO), *Loss*, and *Model*. User can extend it by inheriting the base classes of these modules and implementing the corresponding interfaces; (ii) We try to relieve the pain of coding in C++ so that users can implement algorithms in a “Matlab” style. The code snippet in Fig. 1 shows an example to implement the core function of the “ALMA” algorithm.

2.3. Comparisons

Due to space limitation, we only demonstrate that: (1) the online learning algorithms quickly reach comparable test accuracy compared to L2-SVM in LIBLINEAR [14] and VW²; (2) the sparse online learning methods can select meaningful features compared to L1-SVM in LIBLINEAR and L1-SGD in VW. According to Table 2, SOL provides a wide variety of algorithms that can achieve comparable test accuracies as LIBLINEAR and VW, while the training time is significantly less than LIBLINEAR. VW is also an efficient

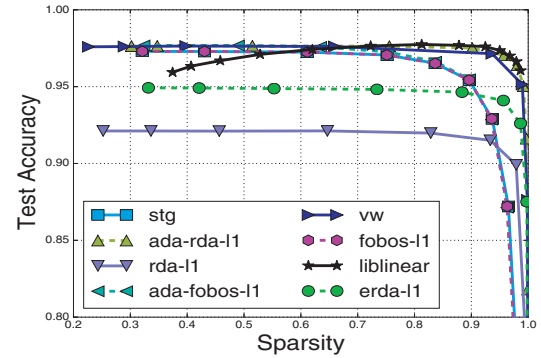


Fig. 2. Comparison of sparse online learning algorithms.

and effective online learning tool, but may not be a comprehensive platform for researchers due to its limited number of algorithms and somewhat complicated designs. Fig. 2 shows how the test accuracy varies with model sparsity. L1-SVM does not work well in low sparsity due to inappropriate regularization. According to the curves, the Ada-RDA-L1 algorithm achieves the best test accuracy for almost all model sparsity values. Clearly, SOL is a highly efficient and effective online learning toolbox. More empirical results on other datasets can be found at <https://github.com/libol/sol/wiki/Example>.

2.4. Illustrative examples

Illustrative examples of SOL can be found at: <https://github.com/libol/sol/wiki/Example>.

3. Conclusion

SOL is an easy-to-use open-source package of scalable online learning algorithms for large-scale online classification tasks. SOL enjoys high efficiency and scalability in practice, particularly when dealing with high-dimensional data. In the era of big data, SOL is not only a sharp knife for machine learning practitioners in learning with massive high-dimensional data, but also a comprehensive research platform for online learning researchers.

Required metadata

Current executable software version

Ancillary data table required for sub version of the executable software: (x.1, x.2 etc.) kindly replace examples in right column with the correct information about your executables, and leave the left column as it is.

² https://github.com/JohnLangford/vowpal_wabbit. VW is another OL tool with only a few algorithms.

Current code version

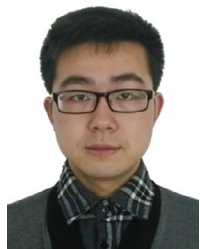
Ancillary data table required for subversion of the codebase. Kindly replace examples in right column with the correct information about your current code, and leave the left column as it is.

Acknowledgments

This research is supported by the National Research Foundation, Prime Minister Office, Singapore under its International Research Centres in Singapore Funding Initiative. This work was done when the first author visited Dr Hoi's group.

References

- [1] S.C. Hoi, J. Wang, P. Zhao, Libol: a library for online learning algorithms, *J. Mach. Learn. Res.* 15 (1) (2014) 495–499.
- [2] J. Langford, L. Li, T. Zhang, Sparse online learning via truncated gradient, *J. Mach. Learn. Res.* 10 (2009) 777–801.
- [3] L. Xiao, Dual averaging methods for regularized stochastic learning and online optimization, *J. Mach. Learn. Res.* 9999 (2010) 2543–2596.
- [4] K. Crammer, A. Kulesza, M. Dredze, Adaptive regularization of weight vectors, *Mach. Learn.* (2009) 1–33.
- [5] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain., *Psychol. Rev.* 65 (6) (1958) 386.
- [6] M. Zinkevich, Online convex programming and generalized infinitesimal gradient ascent (2003). in Ref. [6].
- [7] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, Y. Singer, Online passive-aggressive algorithms, *J. Mach. Learn. Res.* 7 (2006) 551–585.
- [8] C. Gentile, A new approximate maximal margin classification algorithm, *J. Mach. Learn. Res.* 2 (2002) 213–242.
- [9] N. Cesa-Bianchi, A. Conconi, C. Gentile, A second-order perceptron algorithm, *SIAM J. Comput.* 34 (3) (2005) 640–668.
- [10] M. Dredze, K. Crammer, F. Pereira, Confidence-weighted linear classification, in: *Proceedings of the 25th international conference on Machine learning, ACM, 2008*, pp. 264–271.
- [11] K. Crammer, M. Dredze, F. Pereira, Exact convex confidence-weighted learning, in: *Proceedings of the Advances in Neural Information Processing Systems, 2008*, pp. 345–352.
- [12] J. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, *J. Mach. Learn. Res.* 12 (2011) 2121–2159.
- [13] J. Duchi, Y. Singer, Efficient online and batch learning using forward backward splitting, *J. Mach. Learn. Res.* 10 (2009) 2899–2934.
- [14] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, C.-J. Lin, Liblinear: a library for large linear classification, *J. Mach. Learn. Res.* 9 (2008) 1871–1874.



Yue Wu received his B.E. degree in 2012 from the University of Science and Technology of China (USTC), Hefei, China. He is currently a Ph.D. candidate in the Department of Electronic Engineering (EELS), USTC. His research interests include multimedia, computer vision, machine learning, and data mining.



Steven C. H. Hoi is currently an Associate Professor of the School of Information Systems, Singapore Management University, Singapore. Prior to joining SMU, he was Associate Professor with Nanyang Technological University, Singapore. He received his Bachelor degree from Tsinghua University, P.R. China, in 2002, and his Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, in 2006. His research interests are machine learning and data mining and their applications to multimedia information retrieval (image and video retrieval), social media and web mining, and computational finance, etc, and he has published over 150 refereed papers in top conferences and journals in these related areas. He has served as Associate Editor-in-Chief for *Neurocomputing Journal*, general co-chair for ACM SIGMM Workshops on Social Media (WSM'09, WSM'10, WSM'11), program co-chair for the fourth Asian Conference on Machine Learning (ACML'12), book editor for *Social Media Modeling and Computing*, guest editor for *ACM Transactions on Intelligent Systems and Technology (ACM TIST)*, technical PC member for many international conferences, and external reviewer for many top journals and worldwide funding agencies, including NSF in US and RGC in Hong Kong.



Chenghao Liu received his B.Sc. degree in Computer Science and Technology College, Zhejiang University, China in 2011. He is currently pursuing the Ph.D. degree from the College of Computer Science, Zhejiang University, China. His current research interests include machine learning and data mining.



Jing LU is a Ph.D. student in the School of Information Systems (SIS), Singapore Management University (SMU), Singapore. Prior to joining SMU, She received her Bachelor's Degree in Honour School, Harbin Institute of Technology, China in 2012 and worked as a Project Officer in School of Computer Science, Nanyang Technological University, Singapore 2012–2014. During her past Ph.D. study, she has been dedicated to her research area of online learning for addressing the emerging challenges of big data analytics particularly for dealing with real-time data stream analytics. She has published several research papers as the first authors in top tier journals and high-impact conferences. Most of her research publications addressed the key open challenges in the area of machine learning and big data analytics fields.



Doyen SAHOO is a Ph.D. Candidate in School of Information Systems, Singapore Management University. He is supervised by Associate Professor Steven C.H. HOI. His primary research topic is Online Learning with nonlinear models. He works on theoretical aspects of machine learning with focus on Online Learning, Deep Learning and Multiple Kernel Learning. He also works on applications of machine learning to Portfolio Optimization and Cyber-Security. Prior to starting PhD, Doyen completed his B.Eng. in Computer Science from Nanyang Technological University.



Nenghai Yu received the B.S. degree from Nanjing University of Posts and Telecommunications, Nanjing, China, in 1987, the M.E. degree from Tsinghua University, Beijing, China, in 1992, and the Ph.D. degree from University of Science and Technology of China (USTC), Hefei, China, in 2004. He has been on the faculty of the Department of Electronic Engineering and Information Science (EELS), University of Science and Technology of China since 1992, where he is currently a professor. He is the executive director of the Department of EELS, and the director of the Information Processing Center at USTC. His research interests include multimedia security, multimedia information retrieval, video processing and information hiding. He has

authored or co-authored over 130 papers in journals and international conferences. He has been responsible for many national research projects. Prof. Yu and his research group won the Excellent Person Award and the Excellent Collectivity Award simultaneously from the National Hi-tech Development Project of China in 2004. He was the co-author of the Best Paper Candidate at ACM Multimedia 2008.