Journal of Machine Learning Research 15 (2014) 495-499 Submitted 9/13; Revised 12/13; Published 2/14

LIBOL: A Library for Online Learning Algorithms

Steven C.H. Hoi CHHOI@NTU.EDU.SG
Jialei Wang JL.WANG@NTU.EDU.SG
Peilin Zhao ZHAO0106@NTU.EDU.SG
School of Computer Engineering
Nanyang Technological University
Singapore 639798

Editor: Mark Reid

Abstract

LIBOL is an open-source library for large-scale online learning, which consists of a large family of efficient and scalable state-of-the-art online learning algorithms for large-scale online classification tasks. We have offered easy-to-use command-line tools and examples for users and developers, and also have made comprehensive documents available for both beginners and advanced users. LIBOL is not only a machine learning toolbox, but also a comprehensive experimental platform for conducting online learning research.

Keywords: online learning, massive-scale classification, big data analytics

1. Introduction

Online learning represents an important family of efficient and scalable machine learning algorithms for large-scale applications. In general, online learning algorithms are fast, simple, and often make few statistical assumptions, making them applicable to a wide range of applications. Online learning has been actively studied in several communities, including machine learning, statistics, and artificial intelligence. Over the past years, a variety of online learning algorithms have been proposed, but so far there is very few comprehensive library which includes most of the state-of-the-art algorithms for researchers to make easy side-by-side comparisons and for developers to explore their various applications.

In this work, we develop LIBOL as an easy-to-use online learning tool that consists of a large family of classical and recent state-of-the-art online learning algorithms for large-scale online classification tasks. In contrast to many existing software for large-scale data classification, LIBOL enjoys significant advantages for massive-scale data classification in the era of big data nowadays, especially in efficiency, scalability, parallelization, and adaptability. Our goal is to make LIBOL not only a useful machine learning tool for practical users, but also a comprehensive experimental platform for machine learning researchers to conduct online learning research. The LIBOL software is available at `http://libol.stevenhoi.org/`. A more comprehensive and up-to-date documentation for the latest software is available at `http://libol.stevenhoi.org/LIBOL_manual.pdf`.

2. A Family of Online Learning Algorithms for Linear Classification

Online learning operates on a sequence of data examples with time stamps. At each step t, the learner receives an incoming example $\mathbf{x}_t \in \mathcal{X}$ in a d-dimensional vector space, that is, $\mathcal{X} = \mathbb{R}^d$. It first attempts to predict the class label of the incoming instance, $\hat{y}_t = \mathsf{sgn}(f(\mathbf{x}_t; \mathbf{w}_t)) = \mathsf{sgn}(\mathbf{w}_t \cdot \mathbf{x}_t) \in \mathcal{Y}$, and $\mathcal{Y} = \{-1, +1\}$ for binary classification tasks. After making the prediction, the true label $y_t \in \mathcal{Y}$ is revealed, and the learners then computes the loss $\ell(y_t, \hat{y}_t)$ based on some criterion to measure the difference between the learner's prediction and the revealed true label y_t. Based on the result of the loss, the learner finally decides when and how to update the classification model at the end of each learning step. The following algorithmic framework gives an overview of most online learning algorithms[1] for linear classification, where $\Delta(\mathbf{w}_t; (\mathbf{x}_t, y_t))$ denotes the update of the classification models. Different online learning algorithms in general are distinguished in terms of different definitions and designs of the loss function $\ell(\cdot)$ and their various updating functions $\Delta(\cdot)$.

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

Initialize: $\mathbf{w}_1 = 0$
for $t = 1, 2, \ldots, T$ **do**
  The learner receives an incoming instance: $\mathbf{x}_t \in \mathcal{X}$;
  The learner predicts the class label: $\hat{y}_t = \mathsf{sgn}(f(\mathbf{x}_t; \mathbf{w}_t))$;
  The true class label is revealed from the environment: $y_t \in \mathcal{Y}$;
  The learner calculates the suffered loss: $\ell(\mathbf{w}_t; (\mathbf{x}_t, y_t))$;
  if $\ell(\mathbf{w}_t; (\mathbf{x}_t, y_t)) > 0$ **then**
    The learner updates the classification model:
    $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + \Delta(\mathbf{w}_t; (\mathbf{x}_t, y_t))$;
  end
end

The goal of our work is to implement a large family of diverse online learning methods in literature to facilitate research and development of online learning techniques to real-world applications. In particular, this software consists of 29 different online learning algorithms and variants for both binary and multiclass classification tasks. In general, they can be grouped into two major categories: (i) first order learning (Rosenblatt, 1958; Crammer et al., 2006), and (ii) second order learning (Dredze et al., 2008; Wang et al., 2012a; Yang et al., 2009). Examples of online learning algorithms in the first order learning category include the following list of classical and popular algorithms:

- Perceptron: the classical online learning algorithm (Rosenblatt, 1958);
- ALMA: A New Approximate Maximal Margin Classification Algorithm (Gentile, 2001);
- ROMMA: the relaxed online maximum margin algorithms (Li and Long, 2002);
- OGD: the Online Gradient Descent (OGD) algorithms (Zinkevich, 2003);
- PA: the Passive Aggressive (PA) online learning algorithms (Crammer et al., 2006).

1. Note that second order online learning algorithms follow a slightly different procedure.

To improve the efficacy of first order learning methods, recent years have witnessed the emerging active studies of second order online learning algorithms. One family of recent second order algorithms (Dredze et al., 2008) assume the weight vector follows a Gaussian distribution $\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ with mean vector $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance matrix $\Sigma \in \mathbb{R}^{d\times d}$. The model parameters, including both Σ and $\boldsymbol{\mu}$ are updated in the online learning process. Examples of the second order online learning algorithms include the following:

- SOP: the Second Order Perceptron (SOP) algorithm (Cesa-Bianchi et al., 2005);
- CW: the Confidence-Weighted (CW) learning algorithm (Dredze et al., 2008);
- IELLIP: online learning algorithms by improved ellipsoid method (Yang et al., 2009);
- AROW: the Adaptive Regularization of Weight Vectors (Crammer et al., 2009);
- NAROW: New variant of Adaptive Regularization (Orabona and Crammer, 2010);
- NHERD: the Normal Herding method via Gaussian Herding (Crammer and Lee, 2010)
- SCW: the recently proposed Soft Confidence Weighted algorithms (Wang et al., 2012a).

3. The Software Package

The current version (V0.3.0) of LIBOL implements a large family of online learning algorithms and their variants, including 16 algorithms for binary classification, and 13 algorithms for multiclass classification. The package includes the MATLAB library and command-line tools for both online binary and multiclass classification tasks. In addition to MATLAB implementation, we also provide C/C++ implementation for the core functions. The data formats used by this software are compatible with popular machine learning and data mining packages, such as LIBSVM, SVM-light, and WEKA, etc.

3.1 Practical Usage

To illustrate the online learning procedure, we take two data sets from the LIBSVM website, including one small data set "svmguide3" with 1243 instances and one large data set "ijcnn1" with 141,691 instances. In the following example, we use the default "Perceptron" algorithm to demo the usage of LIBOL for a binary classification ('bc') task:

```
$ demo('bc','Perceptron','svmguide3')
```

The results output by the above command are summarized as follows:

Algorithm:	mistake rate	nb of updates	cpu time (seconds)
Perceptron	0.3318 +/- 0.0118	412.45 +/- 14.66	0.0516 +/- 0.0008

To ease researchers to run a full set of experiments for side-by-side comparisons of different algorithms, we offer a very easy-to-use example program as follows:

```
$ run_experiment('bc','svmguide3')
```

The above command will run side-by-side comparison of varied online learning algorithms on the given data set fully automatically, including all the parameter settings and selection. The full set of experimental results will be generated by the library automatically, as shown in Table 1 and Figure 1. This library provides a fairly easy-to-use testbed to facilitate online learning researchers to develop their new algorithms and conduct side-by-side comparisons with the state-of-the-art algorithms with the minimal efforts.

Data Set:	svmguide3 (#samples=1243,#dimensions=36)			ijcnn1 (#samples=141,691,#dimensions=22)		
Algorithm	mistake	# updates	time (s)	mistake	# updates	time (s)
Perceptron	0.332 ± 0.012	412.4 ± 14.7	0.052 ± 0.002	0.106 ± 0.000	15059.9 ± 65.1	5.668 ± 0.064
ROMMA	0.329 ± 0.019	409.3 ± 23.2	0.056 ± 0.001	0.101 ± 0.001	14284.2 ± 89.8	5.823 ± 0.111
aROMMA	0.328 ± 0.018	500.1 ± 29.1	0.055 ± 0.001	0.101 ± 0.001	14776.0 ± 101.8	5.665 ± 0.062
ALMA	0.230 ± 0.006	592.9 ± 6.6	0.062 ± 0.001	0.071 ± 0.000	21474.0 ± 80.4	6.662 ± 0.127
OGD	0.237 ± 0.003	636.5 ± 4.2	0.064 ± 0.002	0.095 ± 0.000	27465.8 ± 31.1	6.369 ± 0.107
PA	0.318 ± 0.013	721.1 ± 18.3	0.060 ± 0.001	0.102 ± 0.001	33847.9 ± 135.2	5.955 ± 0.091
PA1	0.236 ± 0.002	763.4 ± 11.5	0.064 ± 0.001	0.077 ± 0.000	28376.3 ± 84.2	6.352 ± 0.109
PA2	0.253 ± 0.007	1131.5 ± 15.6	0.069 ± 0.001	0.081 ± 0.000	61093.8 ± 199.3	6.876 ± 0.114
SOP	0.297 ± 0.012	369.1 ± 14.8	0.095 ± 0.002	0.102 ± 0.001	14470.7 ± 81.3	10.616 ± 0.096
IELLIP	0.332 ± 0.013	412.7 ± 16.1	0.081 ± 0.002	0.119 ± 0.001	16876.8 ± 72.9	8.079 ± 0.082
CW	0.299 ± 0.011	704.6 ± 19.1	0.118 ± 0.002	0.093 ± 0.001	30648.3 ± 166.2	9.499 ± 0.110
NHERD	0.217 ± 0.007	1150.5 ± 27.7	0.096 ± 0.002	0.084 ± 0.001	86660.7 ± 2692.6	9.735 ± 0.133
AROW	0.222 ± 0.004	1219.5 ± 8.1	0.112 ± 0.002	0.082 ± 0.000	74247.1 ± 846.3	10.164 ± 0.069
NAROW	0.276 ± 0.042	1193.8 ± 23.2	0.118 ± 0.002	0.095 ± 0.008	103843.6 ± 8841.3	12.027 ± 0.467
SCW	0.206 ± 0.004	593.4 ± 13.9	0.085 ± 0.002	0.060 ± 0.002	11077.3 ± 678.3	7.921 ± 0.144
SCW2	0.212 ± 0.009	802.0 ± 73.2	0.092 ± 0.002	0.070 ± 0.001	30833.8 ± 2116.8	8.681 ± 0.150

Table 1: Comparison of a variety of online learning algorithms on two data sets.

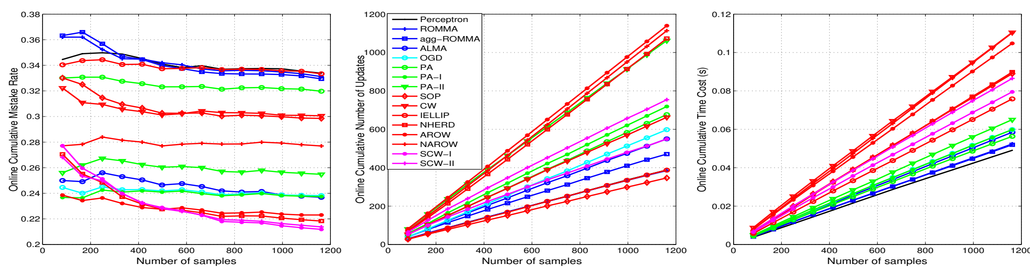

Figure 1: Comparison of a variety of online learning algorithms on data set “svmguide3”.

3.2 Documentation and Design

The LIBOL package comes with comprehensive documentation. The README file describes the setup and usage. Users can read the “Quick Start” section to begin shortly. All the functions and related data structures are explained in detail. If the README file does not give the information users want, they can check the online FAQ. In addition to software documentation, theoretical properties of the algorithms and comparisons can be found in Wang et al. (2012a). The authors are also willing to answer any further questions.

The design principle is to keep the package simple, easy to read and extend. All codes follow the MATALB standards with core functions implemented in C/C++. It generally needs no external libraries, except for the support of popular data formats, such as LIBSVM and WEKA data sets for which existing libraries are included. LIBOL is written in a modular way. All the online learning algorithms can be called via the uniform “ol_train()” function by setting proper options. One can easily develop a new algorithm and make side-by-side comparisons with the existing ones in the package. Our goal is to make LIBOL not only a machine learning tool, but also an experimental platform for online learning research.

4. Conclusion

LIBOL is an easy-to-use open-source package for online learning research and development. The current version of LIBOL includes a large number of online learning algorithms for online classification tasks. LIBOL is still being improved by feedback from practical users and new research results (Zhao et al., 2011a,b; Wang et al., 2012b; Hoi et al., 2013a,b). We hope to make LIBOL not only a useful machine learning tool, but also an ideal research platform for conducting online learning research. The ultimate goal is to make easy learning with massive data streams for tackling the grand challenge of big data analytics.

References

Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. A second-order perceptron algorithm. *SIAM J. Comput.*, 34(3):640–668, 2005.

Koby Crammer and Daniel D. Lee. Learning via gaussian herding. In *NIPS*, pages 451–459, 2010.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006.

Koby Crammer, Alex Kulesza, and Mark Dredze. Adaptive regularization of weight vectors. In *NIPS*, pages 345–352, 2009.

Mark Dredze, Koby Crammer, and Fernando Pereira. Confidence-weighted linear classification. In *ICML*, pages 264–271, 2008.

Claudio Gentile. A new approximate maximal margin classification algorithm. *Journal of Machine Learning Research*, 2:213–242, 2001.

Steven C. H. Hoi, Rong Jin, Peilin Zhao, and Tianbao Yang. Online multiple kernel classification. *Machine Learning*, 90(2):289–316, 2013a.

Steven C. H. Hoi, Jialei Wang, Peilin Zhao, Jinfeng Zhuang, and Zhi-Yong Liu. Large scale online kernel classification. In *IJCAI*, 2013b.

Yi Li and Philip M. Long. The relaxed online maximum margin algorithm. *Machine Learning*, 46(1-3):361–387, 2002.

Francesco Orabona and Koby Crammer. New adaptive algorithms for online classification. In *NIPS*, pages 1840–1848, 2010.

Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psych. Rev.*, 7:551–585, 1958.

Jialei Wang, Peilin Zhao, and Steven C. H. Hoi. Exact soft confidence-weighted learning. *ICML*, 2012a.

Jialei Wang, Peilin Zhao, and Steven CH Hoi. Cost-sensitive online classification. In *IEEE 12th International Conference on Data Mining (ICDM)*, pages 1140–1145. IEEE, 2012b.

Liu Yang, Rong Jin, and Jieping Ye. Online learning by ellipsoid method. In *ICML*, page 145, 2009.

Peilin Zhao, Steven C. H. Hoi, and Rong Jin. Double updating online learning. *Journal of Machine Learning Research*, 12:1587–1615, 2011a.

Peilin Zhao, Steven C. H. Hoi, Rong Jin, and Tianbao Yang. Online auc maximization. In *ICML*, pages 233–240, 2011b.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.