Nanyang Technological University 1 (2013) 1-5 Submitted 4/00; Published 10/00

LIBOL: A Library for Online Learning Algorithms

Steven C.H. Hoi CHHOI@NTU.EDU.SG
Jialei Wang JL.WANG@NTU.EDU.SG
Peilin Zhao PLZHAO@NTU.EDU.SG
School of Computer Engineering
Nanyang Technological University
Singapore 639798
Last modified: July 27, 2013

Editor:

Abstract

LIBOL is an open-source library for large-scale online classification, 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. We also have made comprehensive documents available for both beginners and advanced users. LIBOL is not only a machine learning tool, but also a comprehensive experimental platform for conducting online learning research.

Keywords: Online Learning, Massive-scale Classification, Linear Classification

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 a large family of existing 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 classification in the era of big data nowadays, especially in efficiency, scalability, parallelization, and adaptability. The software is available at `http://libol.stevenhoi.org/`.

This article is organized as follows. Sections 2 discusses the major list of online learning algorithms implemented in this library. Section 3 describes the usage, design and implementation of LIBOL. Section 4 gives our concluding remarks.

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, i.e., $\mathcal{X} = \mathbb{R}^d$. 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: w_1 = 0
for t = 1, 2, ..., T do
    The learner receives an incoming instance: x_t ∈ X;
    The learner predicts the class label: ŷ_t = sgn(f(x_t; w_t));
    The true class label is revealed from the environment: y_t ∈ Y;
    The learner calculates the suffered loss: ℓ(w_t; (x_t, y_t));
    if ℓ(w_t; (x_t, y_t)) > 0 then
        The learner updates the classification model:
        w_{t+1} ← w_t + Δ(w_t; (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 15 different online learning algorithms and their variants. 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., 2012; 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 maxiumu margin algorithms (Li and Long, 2002);
- OGD: the Online Gradient Descent (OGD) algorithms (Zinkevich, 2003);
- PA: Passive Aggressive (PA) algorithms (Crammer et al., 2006), one of state-of-the-art first order online learning algorithms;

1. Except that SOP in (Cesa-Bianchi et al., 2005) follows a slightly different procedure.

In recent years, to improve the first order learning methods, the second order online learning algorithms typically 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 the mean vector and the covariance matrix are updated in the online learning process. Example 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., 2012).

3. The Software Package

The current version (V0.2.0) of LIBOL package implements a large family of online learning algorithms and their variants, including 16 algorithms for binary classification, and 13 algorithms for multi-class classification. The package includes the MATALB 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 package are compatible to some 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	0.3296 +/- 0.0103	409.75 +/- 12.80	0.0458 +/- 0.0006

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 auotmatically, including all the parameter settings and selection.

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

Dataset:	svmguide3 (#samples=1243,#dimensions=36)			ijcnn1 (#samples=141,691,#dimensions=22)		
Algorithm	mistake	# updates	time (s)	mistake	# updates	time (s)
Perceptron	0.330 ± 0.010	409.8 ± 12.8	0.045 ± 0.000	0.106 ± 0.001	15052.9 ± 71.2	5.452 ± 0.128
ROMMA	0.338 ± 0.013	419.9 ± 16.4	0.049 ± 0.000	0.101 ± 0.001	14291.8 ± 72.2	5.720 ± 0.160
aROMMA	0.333 ± 0.014	516.3 ± 24.1	0.053 ± 0.001	0.101 ± 0.001	14806.6 ± 98.5	5.542 ± 0.194
ALMA	0.237 ± 0.007	586.4 ± 10.0	0.059 ± 0.001	0.072 ± 0.000	21753.0 ± 58.7	6.671 ± 0.165
OGD	0.238 ± 0.003	637.5 ± 3.3	0.060 ± 0.001	0.095 ± 0.000	27449.6 ± 45.1	6.692 ± 0.174
PA1	0.236 ± 0.002	763.9 ± 9.9	0.063 ± 0.001	0.077 ± 0.000	28399.0 ± 83.5	6.766 ± 0.230
PA2	0.254 ± 0.005	1134.1± 13.3	0.071 ± 0.001	0.081 ± 0.000	61085.9 ± 154.8	18.961 ± 0.895
SOP	0.294 ± 0.009	365.9 ± 11.7	0.104 ± 0.001	0.102 ± 0.001	14487.8 ± 86.0	11.890 ± 0.348
IELLIP	0.317 ± 0.009	393.4 ± 11.4	0.074 ± 0.002	0.119 ± 0.001	16812.3 ± 135.2	7.978 ± 0.249
CW	0.295 ± 0.008	699.9 ± 13.3	0.094 ± 0.001	0.093 ± 0.001	30678.0 ± 146.9	11.181 ± 0.394
NHERD	0.224 ± 0.013	1152.1± 21.9	0.102 ± 0.001	0.084 ± 0.001	84878.5 ± 4007.8	36.555 ± 3.098
AROW	0.225 ± 0.005	1219.5± 6.5	0.127 ± 0.001	0.082 ± 0.000	73563.1 ± 1318.8	29.979 ± 1.460
NAROW	0.267 ± 0.032	1181.5± 39.1	0.133 ± 0.003	0.097 ± 0.013	103203.7 ± 8566.6	53.171 ± 7.636
SCW	0.214 ± 0.006	583.9 ± 14.9	0.084 ± 0.001	0.058 ± 0.002	10829.3 ± 524.0	7.508 ± 0.262
SCW2	0.215 ± 0.008	784.0 ± 70.5	0.093 ± 0.003	0.070 ± 0.001	30965.5 ± 2433.9	9.518 ± 0.447

The full set of experimental results will be generated by the library automatially, as shown in Table 1 and Figure 1. This library provides a fairly easy-to-use testbed to faciliate 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.

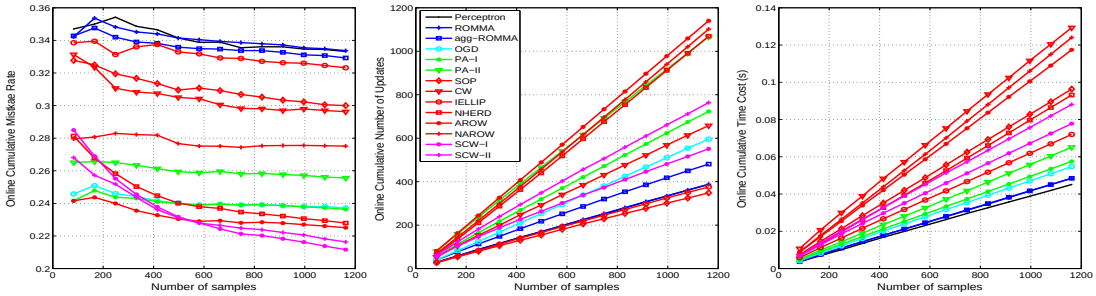

Figure 1: Comparison of a variety of online learning algorithms on dataset "svmguide3".

3.2 Documentation

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. (2012). The authors are also willing to answer any further questions.

3.3 Design

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 datasets for which existing libraries are included. All the online learning algorithms can be called via the uniform "ol_train()" function by setting proper options. In general, LIBOL is written in a modular way, in which one can easily develop a new algorithm and make side-by-side comparisons with the existing ones in the package. Thus, LIBOL is not only a machine learning tool, but also an experimental platform for online learning research.

4. Conclusions

LIBOL is an easy-to-use open source package for efficient and scalable on-line linear classification. It is currently one of comprehensive online learning software packages that include the largest number of diverse online learning algorithms for online classification. LIBOL is still being improved by improvements from practical users and new research results (Zhao et al., 2011a,b). The ultimate goal is to make easy learning with massive-scale data streams to tackle the emerging grand challenge of big data mining.

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.

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*, 2012.

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.