Journal of Machine Learning Research 1 (2015) 1-5 Submitted 4/00; Published 10/00

OLPS: A Toolbox for On-Line Portfolio Selection

Bin Li BINLI.WHU@WHU.EDU.CN
Economics and Management School, Wuhan University, Wuhan, P.R. China 430072
Doyen Sahoo DOYEN.2014@PHDIS.SMU.EDU.SG
Steven C. H. Hoi CHHOI@SMU.EDU.SG
School of Information Systems, Singapore Management University, Singapore 178902

Editor:

Abstract

On-line portfolio selection is a practical financial engineering problem, which aims to sequentially allocate capital among a set of assets in order to maximize long-term return. In recent years, a variety of machine learning algorithms have been proposed to address this challenging problem, but no comprehensive open-source toolbox has been released for various reasons. This article presents the first open-source toolbox for "On-Line Portfolio Selection" (OLPS), which implements a collection of classical and state-of-the-art strategies powered by machine learning algorithms. We hope that OLPS can facilitate the development of new learning methods and enable the performance benchmarking and comparisons of different strategies. OLPS is an open-source project released under Apache License (version 2.0), which is available at `https://github.com/OLPS/` or `http://OLPS.stevenhoi.org/`.

Keywords: On-line portfolio selection, online learning, trading system, simulation.

1. Introduction

In recent years, machine learning has been applied to various applications in finance (Györfi et al., 2012), including On-line Portfolio Selection, which aims to sequentially allocate capital among a set of assets, such that the investment return can be maximized in the long run (Kelly, 1956). It has attracted increasing attention from both academia and industry, and several machine learning algorithms have been proposed (Li and Hoi, 2014), including traditional algorithms (Cover, 1991; Helmbold et al., 1998; Agarwal et al., 2006; Borodin et al., 2004; Györfi et al., 2006, 2008), and recent state-of-the-art online learning algorithms (Li et al., 2011, 2012, 2013, 2015). Unlike other application domains in machine learning where various open-source packages are available, very few open-source toolkits[1] exist for on-line portfolio selection, primarily due to the confidential nature of financial industry. Consequently, it is difficult for researchers to evaluate new algorithms for comprehensive comparisons with existing ones.

This article introduces an open-source toolkit named "On-Line Portfolio Selection" (OLPS), which consists of a family of classical and state-of-the-art on-line portfolio selection algorithms. To the best of our knowledge, OLPS is the first comprehensive open-source package for this problem, which includes various strategies, and a set of preprocessing, postprocessing and visualization tools

[1] Some repositories contain public datasets, such as `http://www.cs.technion.ac.il/~rani/portfolios` and `http://www.szit.bme.hu/~oti/portfolio`, but no public code was released.

in an integrated platform. The open-source nature of OLPS makes it easy for a third party to develop new algorithms, and facilitates the comparisons with many built-in algorithms on real datasets.

2. Overview and Implementation

OLPS implements a framework for backtesting various algorithms for on-line portfolio selection in both Matlab and Octave (the GUI is only available in Matlab) under Windows, Linux, and Mac OS. Figure 1 gives an overview of the OLPS toolkit with three main modules: (i) data preprocessing: it loads a dataset and initializes the backtesting environments, e.g., preparing log file handles, etc; (ii) algorithms: it calls selected algorithms, and simulates the trading process based on the chosen data from the first module; and (iii) post-processing: it analyzes statistical significance of the results from the second module, e.g., some risk-adjusted returns.

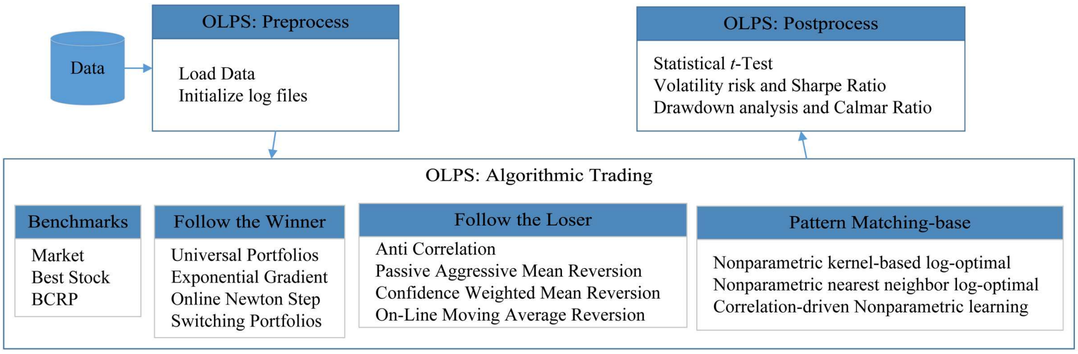

Data
OLPS: Preprocess
Load Data
Initialize log files
OLPS: Postprocess
Statistical *t*-Test
Volatility risk and Sharpe Ratio
Drawdown analysis and Calmar Ratio
OLPS: Algorithmic Trading
Benchmarks
Market
Best Stock
BCRP
Follow the Winner
Universal Portfolios
Exponential Gradient
Online Newton Step
Switching Portfolios
Follow the Loser
Anti Correlation
Passive Aggressive Mean Reversion
Confidence Weighted Mean Reversion
On-Line Moving Average Reversion
Pattern Matching-base
Nonparametric kernel-based log-optimal
Nonparametric nearest neighbor log-optimal
Correlation-driven Nonparametric learning

Figure 1: Structure of the OLPS toolbox.

2.1 Pre-processing

Datasets	Markets	Regions	Time Frames	# Periods	# Assets	File Names (.mat)
NYSE (O)	Stock	US	07/03/1962 - 12/31/1962	5651	36	`nyse-o`
NYSE (N)	Stock	US	01/01/1985 - 06/30/2010	6431	23	`nyse-n`
TSE	Stock	CA	01/04/1994 - 12/31/1998	1259	88	`tse`
SP500	Stock	US	01/02/1998 - 01/31/2003	1276	25	`sp500`
MSCI	Index	Global	04/01/2006 - 03/31/2010	1043	24	`msci`
DJIA	Stock	US	01/14/2001 - 01/14/2003	507	30	`djia`

Table 1: Summary of the six datasets from real markets.

The main functionality of this step is to initialize the trading environment. OLPS supports all types of datasets that Matlab/Octave accepts, and all existing datasets are in MAT-file. A typical dataset contains an $n \times m$ matrix of price relatives, where n denotes the number of trading periods, and m refers to the number of assets. It can be further adapted to incorporate real market data feeds, such that the toolkit can handle real time data and conduct paper trading or real trading[2]. Table 1

[2] For example, Interactive Brokers (`http://www.interactivebrokers.com`) provides free APIs. Paper and real trading both require to implement order submission, while back test does not.

summarizes several representative public datasets included in the toolbox[3]. Users are free to collect up-to-date data from various sources, such as the CRSP database, Quandl.com, etc.

2.2 Algorithmic Trading

This step conducts simulation based on historical market data. Table 2 summarizes the families of algorithms implemented in the OLPS toolbox. In our framework, implementing a new strategy generally requires four files, i.e., an entry file, a run file, a kernel file and an expert file. The entry file extracts parameters and call the corresponding run file. The run file simulates a whole trading process, and calls its kernel file to construct a portfolio for each period, which is used for rebalancing. The kernel file outputs a portfolio, while it facilitates the development of meta algorithms, which combine multiple experts' portfolios outputted by expert files that output one portfolio. In case of only one expert, the kernel file is not necessary and directly enters the expert file. Developing new strategies involves writing a kernel file, which aims to output portfolio.

Categories	Algorithms	Entry Files	References
Baselines	Uniform BAH	`ubah.m`	Li and Hoi (2014)
	Best Stock	`best.m`	Li and Hoi (2014)
	Uniform CRP	`ucrp.m`	Li and Hoi (2014)
	BCRP	`bcrp.m`	Cover (1991)
Follow the Winner	UP	`up.m`	Cover (1991)
	EG	`eg.m`	Helmbold et al. (1998)
	ONS	`ons.m`	Agarwal et al. (2006)
	SP	`sp.m`	Singer (1997)
Follow the Loser	Anticor	`anticor/anticor_anticor.m`	Borodin et al. (2004)
	PAMR	`pamr/pamr_1/pamr_2.m`	Li et al. (2012)
	CWMR	`cwmr_stdev/cwmr_var.m`	Li et al. (2013)
	OLMAR	`olmar1/olmar2.m`	Li et al. (2015)
Pattern Matching based Algorithms	B^K	`bk.m`	Györfi et al. (2006)
	B^{NN}	`bnn.m`	Györfi et al. (2008)
	CORN	`corn/cornu/cornk.m`	Li et al. (2011)
Others	M0	`m0.m`	Borodin et al. (2000)
	T0	`t0.m`	Borodin et al. (2000)

Table 2: Summary of the implemented algorithms.

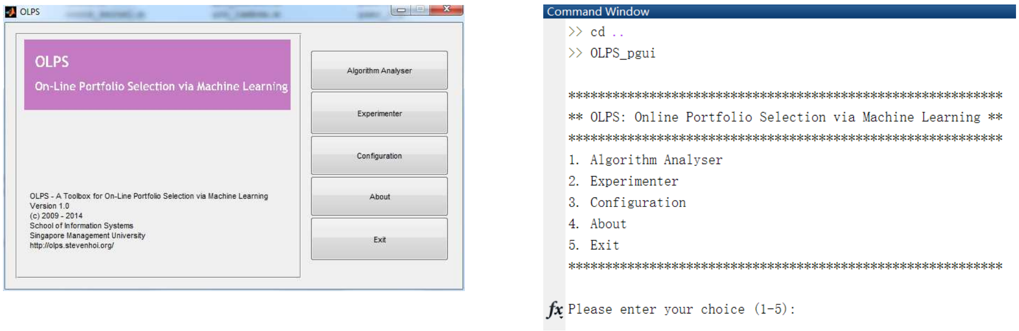
2.3 Post-processing

After the algorithmic trading simulation, the last step is to post-process the portfolio's return series for performance analysis by summarizing the following performance metrics:

- Cumulative Return: the most widely used performance metric in related studies;
- Volatility and Sharpe Ratio: measures volatility risk and related risk-adjusted return;
- Drawdown and Calmar Ratio: measures downside risk and risk-adjusted return;
- T-test statistics (Grinold and Kahn, 1999): tests if a strategy's return is significantly different from the market.

[3] More datasets and their details, including the components, can be found at the project website.

3. Usage



OLPS - A Toolbox for On-Line Portfolio Selection via Machine Learning
Version 1.0
(c) 2009 - 2014
School of Information Systems
Singapore Management University
http://olps.stevenhoi.org/

(a) GUI mode

```
>> cd ..
>> OLPS_pgui

***********************************************************
** OLPS: Online Portfolio Selection via Machine Learning **
***********************************************************
1.  Algorithm Analyser
2.  Experimenter
3.  Configuration
4.  About
5.  Exit
***********************************************************

fx Please enter your choice (1-5):
```

(b) Pseudo GUI mode

Figure 2: GUI mode and Pseudo GUI mode.

We provide three interfaces to run simulations in the toolbox. As shown in Figure 2, the toolbox has two types of Graphical User Interfaces (GUIs), including a GUI implemented in Matlab and a Pseudo GUI in command line, which is thus available in both Matlab and Octave. The other interface is a Command Line Interface (CLI), and is also available in both Matlab and Octave.

While a GUI is straightforward to use, in the CLI mode, we generalize a *manager* function so as to control the running of all algorithms.Suppose we are going to simulate the PAMR algorithm on the NYSE (O) dataset. The commands can be listed as follows:

```
1: >> opts.quiet_mode=1; opts.display_interval=500; opts.progress=0;
2: >> opts.analyze_mode=1; opts.log_mode=1; opts.mat_mode=1;
3: >> manager('pamr', 'nyse-o', {0.5, 0}, opts);
```

Line 1 and 2 set some variables that are used to control the display and file storage. Line 3 calls the manager function to execute the "pamr" strategy on the "nyse-o" dataset with the parameters equaling "{0.5, 0}". During the simulation, the toolbox will output information periodically. After the simulation, the algorithm will analyze the returns and output the cumulative return, risk adjusted returns and statistical test statistics. Corresponding running details and results will be automatically stored in the Log folder.

4. Summary

This article presented OLPS — an open-source On-Line Portfolio Selection toolbox to facilitate the related research in machine learning and computational finance. This is the first open-source project in the area, which not only facilitates researchers to develop new strategies, but also allows them to easily benchmark their performance with existing strategies. Besides, the toolbox supports a large collection of classical and state-of-the-art on-line portfolio selection strategies. The toolbox offers a user-friendly GUI in Matlab, a Pseudo GUI and a CLI mode in both Octave and Matlab. We hope that the open-source nature of the software would encourage researchers to extend the toolkit and share their algorithms through the OLPS platform.

Acknowledgments

This work was partially supported by the Project of NSFC (No. 71401128 and 71471142), the Project of SRF for ROCS, SEM, and Singapore MOE tier 1 project (C220/MSS14C003).

References

A. Agarwal, E. Hazan, S. Kale, and R. E. Schapire. Algorithms for portfolio management based on the newton method. In *Proceedings of ICML*, 2006.

A. Borodin, R. El-Yaniv, and V. Gogan. On the competitive theory and practice of portfolio selection (extended abstract). In *Proceedings of the Latin American Symposium on Theoretical Informatics*, 2000.

A. Borodin, R. El-Yaniv, and V. Gogan. Can we learn to beat the best stock. *JAIR*, 21:579–594, 2004.

T. M. Cover. Universal portfolios. *Mathematical Finance*, 1(1):1–29, 1991.

R. Grinold and R. Kahn. *Active Portfolio Management: A Quantitative Approach for Producing Superior Returns and Controlling Risk*. New York : McGraw-Hill, New York, 1999.

L. Györfi, G. Lugosi, and F. Udina. Nonparametric kernel-based sequential investment strategies. *Mathematical Finance*, 16(2):337–357, 2006.

L. Györfi, F. Udina, and H. Walk. Nonparametric nearest neighbor based empirical portfolio selection strategies. *Statistics and Decisions*, 26(2):145–157, 2008.

L. Györfi, G. Ottucsák, and H. Walk. *Machine Learning for Financial Engineering*. Singapore : World Scientific, 2012.

D. P. Helmbold, R. E. Schapire, Y. Singer, and M. K. Warmuth. On-line portfolio selection using multiplicative updates. *Mathematical Finance*, 8(4):325–347, 1998.

Jr. Kelly, J. A new interpretation of information rate. *Bell Systems Technical Journal*, 35:917–926, 1956.

B. Li and S. C. H. Hoi. Online portfolio selection: A survey. *ACM CSUR*, 46(3):35, 2014.

B. Li, S. C.H. Hoi, and V. Gopalkrishnan. Corn: Correlation-driven nonparametric learning approach for portfolio selection. *ACM TIST*, 2(3):21:1–21:29, 2011.

B. Li, P. Zhao, S. Hoi, and V. Gopalkrishnan. Pamr: Passive aggressive mean reversion strategy for portfolio selection. *Machine Learning*, 87(2):221–258, 2012.

B. Li, S. C.H. Hoi, P. Zhao, and V. Gopalkrishnan. Confidence weighted mean reversion strategy for online portfolio selection. *ACM TKDD*, 7(1):4:1 – 4:38, 2013.

B. Li, S. C.H. Hoi, D. Sahoo, and Z. Liu. Moving average reversion strategy for on-line portfolio selection. *Artificial Intelligence*, 222:104 – 123, 2015.

Y. Singer. Switching portfolios. *International Journal of Neural Systems*, 8(4):488–495, 1997.