Online Gradient Descent for Online Portfolio Optimization with Transaction Costs

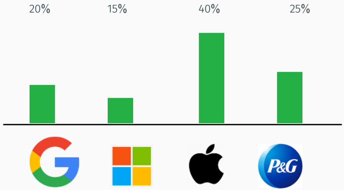
Martino Nicoló Bernasconi de Luca
Supervisor: Francesco Trovó
Co-supervisors: Marcello Restelli, Edoardo Vittori

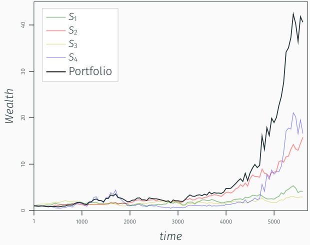
POLITECNICO
MILANO 1863

Table of contents

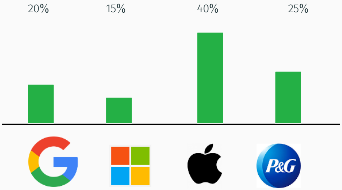
Context

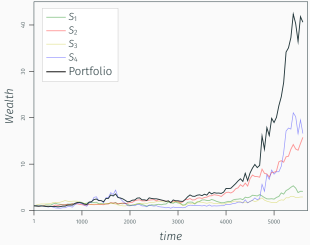
Online Portfolio Optimization





Online Portfolio Optimization

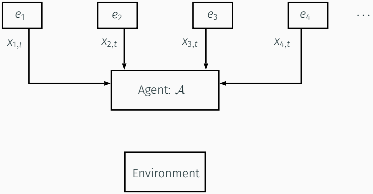




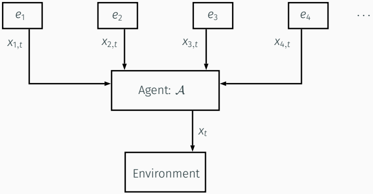
- Modern Portfolio Optimization:
  - Statistical assumptions on the stock dynamics
  - Backward looking
- Online Portfolio Optimization:
  - Adversarial (no assumptions on the distribution)
  - From Online Learning field

Context: Online Learning with Experts Advice

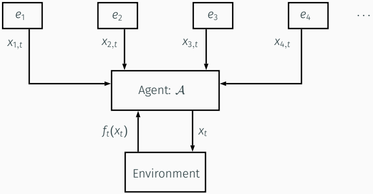
Context: Online Learning with Experts Advice



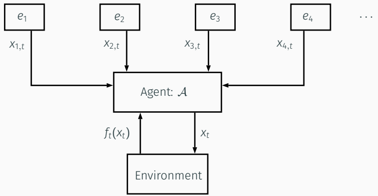
Context: Online Learning with Experts Advice



Context: Online Learning with Experts Advice

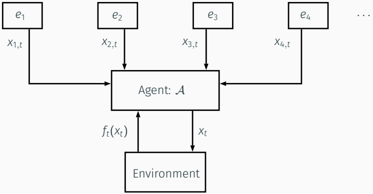


Context: Online Learning with Experts Advice

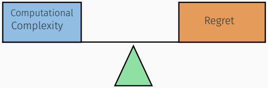


- Regret: $R_T = \sum_{t=1}^{T} f_t(x_t) - \inf_{e \in \mathcal{E}} \sum_{t=1}^{T} f_t(x_{e,t})$
  - No regret: $R_T = o(T)$ for any sequence $f_1, f_2, \ldots$
- Per-round Computational Complexity

Context: Online Learning with Experts Advice



- Regret: $R_T = \sum_{t=1}^{T} f_t(x_t) - \inf_{e \in \mathcal{E}} \sum_{t=1}^{T} f_t(x_{e,t})$
  - No regret: $R_T = o(T)$ for any sequence $f_1, f_2, \ldots$
- Per-round Computational Complexity



Problem Formulation

From Online Learning to Portfolio

- $\mathsf{x}_t \in \Delta_{N-1}$ is the portfolio allocation
- $\mathsf{y}_t = \left(\frac{p_{t,1}}{p_{t-1,1}}, \ldots, \frac{p_{t,N}}{p_{t-1,N}}\right)$ are the returns of the stocks
- $f_t(\mathsf{x}) = -\log(\langle \mathsf{x}, \mathsf{y}_t \rangle)$ is the loss
- $\mathcal{E}$ plays $\mathsf{x}_t = \mathsf{x}$ at each turn (Constant Rebalancing Portfolios).

  $\mathsf{x}^* = \operatorname*{arg\,inf}_{x \in \Delta_{N-1}} \sum_{t=1}^{T} f_t(\mathsf{x})$ is the Best Constant Rebalancing Portfolio

From Online Learning to Portfolio

- $\mathbf{x}_t \in \Delta_{N-1}$ is the portfolio allocation
- $\mathbf{y}_t = \left(\frac{p_{t,1}}{p_{t-1,1}}, \ldots, \frac{p_{t,N}}{p_{t-1,N}}\right)$ are the returns of the stocks
- $f_t(\mathbf{x}) = -\log(\langle \mathbf{x}, \mathbf{y}_t \rangle)$ is the loss
- $\mathcal{E}$ plays $\mathbf{x}_t = \mathbf{x}$ at each turn (Constant Rebalancing Portfolios).
  $\mathbf{x}^* = \arg\inf_{x \in \Delta_{N-1}} \sum_{t=1}^{T} f_t(\mathbf{x})$ is the Best Constant Rebalancing Portfolio

Regret in Online Portfolio Optimization

$$R_T = \log(W_T(\mathbf{x}^*, \ldots, \mathbf{x}^*)/W_T(\mathbf{x}_1, \ldots, \mathbf{x}_T)),$$

where $W_T(\mathbf{x}_1, \ldots, \mathbf{x}_T) = \prod_{t=1}^{T} \langle \mathbf{x}_t, \mathbf{y}_t \rangle$ is the wealth

From Online Learning to Portfolio

- $\mathbf{x}_t \in \Delta_{N-1}$ is the portfolio allocation
- $\mathbf{y}_t = \left(\frac{p_{t,1}}{p_{t-1,1}}, \ldots, \frac{p_{t,N}}{p_{t-1,N}} \right)$ are the returns of the stocks
- $f_t(\mathbf{x}) = -\log(\langle \mathbf{x}, \mathbf{y}_t \rangle)$ is the loss
- $\mathcal{E}$ plays $\mathbf{x}_t = \mathbf{x}$ at each turn (Constant Rebalancing Portfolios).
  $\mathbf{x}^* = \operatorname*{arg\,inf}_{x \in \Delta_{N-1}} \sum_{t=1}^{T} f_t(\mathbf{x})$ is the Best Constant Rebalancing Portfolio

Regret in Online Portfolio Optimization

$$R_T = \log(W_T(\mathbf{x}^*, \ldots, \mathbf{x}^*)/W_T(\mathbf{x}_1, \ldots, \mathbf{x}_T)),$$

where $W_T(\mathbf{x}_1, \ldots, \mathbf{x}_T) = \prod_{t=1}^{T} \langle \mathbf{x}_t, \mathbf{y}_t \rangle$ is the wealth

Limitations: No transaction costs

Adding Trading Costs

Regret in Online Portfolio Optimization with Transaction Costs

$$R_T^C = \underbrace{\log(W_T(\mathbf{x}^*, \ldots, \mathbf{x}^*)/W_T(\mathbf{x}_1, \ldots, \mathbf{x}_T))}_{R_T\text{: standard regret}} + \underbrace{\gamma \sum_{t=1}^{T} ||\mathbf{x}_t - \mathbf{x}_{t-1}||_1}_{C_T\text{: transaction costs}},$$

where γ is the transaction rate

γ is the proportional transaction rate for buying and selling stocks

State of the Art

State of the Art Algorithms

- Online Portfolio Optimization:
  - Online Newton Step (ONS) [Agarwal et al. (2006)]
  - Universal Portfolios (UP) [Cover and Ordentlich (1996)]
- Online Portfolio Optimization with Transaction Costs:
  - Online Lazy Updates (OLU) [Das et al. (2013)]
- Heuristics:
  - Passive Aggressive Mean Reversion (PAMR) [Li et al. (2012)]
  - Online Moving Average Reversion (OLMAR) [Li et al. (2015)]

Proposed Solution

Online Gradient Descent

Algorithm 1 OGD in Online Portfolio Optimization with Transaction Costs

Require: learning rate sequence $\{\eta_1, \ldots, \eta_T\}$

1: Set $\mathbf{x}_1 \leftarrow \frac{1}{N}\mathbf{1}$
2: **for** $t \in \{1, \ldots, T\}$ **do**
3: $\quad \mathbf{z}_{t+1} \leftarrow \mathbf{x}_t + \eta_t \frac{\mathbf{y}_t}{\langle \mathbf{y}_t, \mathbf{x}_t \rangle}$
4: $\quad$ Select Portfolio $\mathbf{x}_{t+1} = \underset{\mathbf{x} \in \Delta_{N-1}}{\arg\inf} ||\mathbf{z}_t - \mathbf{x}||_2^2$
5: $\quad$ Observe $\mathbf{y}_{t+1}$ from the market
6: $\quad$ Get wealth $\log(\langle \mathbf{y}_{t+1}, \mathbf{x}_{t+1} \rangle) - \gamma ||\mathbf{x}_{t+1} - \mathbf{x}_t||_1$
7: **end for**

Regret [Zinkevich (2003)]

$$R_T \leq \left(\frac{1}{K} + \frac{NK\epsilon_u^2}{\epsilon_l^2} \right) \sqrt{T}$$

Total Regret (this work)

$$R_T^C \leq \left[\frac{1}{K} + \frac{NK\epsilon_u}{\epsilon_l} \left(\frac{\epsilon_u}{\epsilon_l} + 2\gamma \right) \right] \sqrt{T}$$

Theoretical and Computational Comparison

Comparison of the theoretical guarantees and computational complexity among the selected algorithms

	OGD	UP	ONS	OLU
R_T	$\mathcal{O}(\sqrt{T})$	$\mathcal{O}(\log T)$	$\mathcal{O}(\log T)$	$\mathcal{O}(\sqrt{T})$
R_T^C	$\mathcal{O}(\sqrt{T})$	-	-	$\mathcal{O}(T)$[1]
Complexity	$\Theta(N)$	$\Theta(T^N)$	$\Theta(N^2)$	$\Theta(N)$

[1]Assuming $\gamma = const.$, if $\gamma \propto \sqrt{T}$ then $R_T^C = \mathcal{O}(\sqrt{T})$

Experiments

Experimental Setting

Datasets				
Name	Market	Year Span	Days	Assets
NYSE(O)	New York Stock Exchange	1962 - 1984	5651	36
TSE	Toronto Stock Exchange	1994 - 1998	1258	88
SP500	Standard Poor's 500	1998 - 2003	1276	25

Performance Metrics:

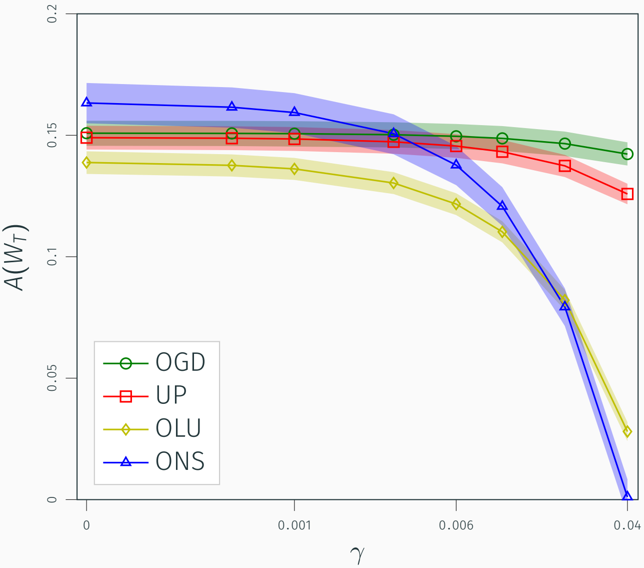
Sample for 100 times, 5 randomly drawn assets

- Average Annual Percentage Yield (APY) defined as

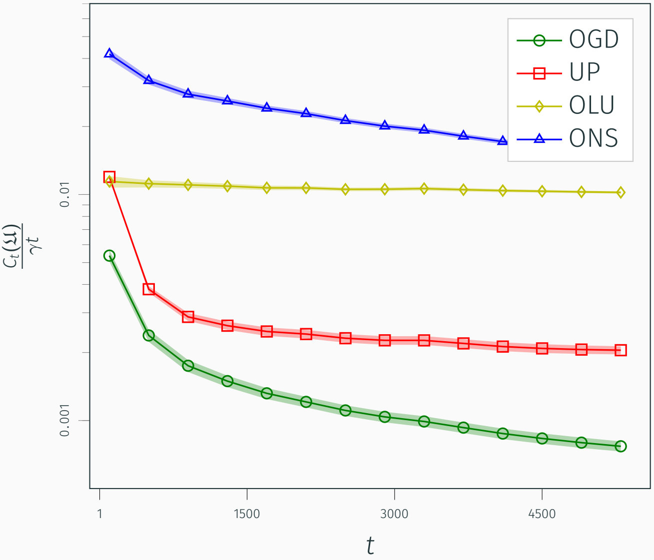
$$A(W_T) = W_T^{250/T} - 1$$

- Normalized per round transaction costs $\frac{C_t(\mathfrak{U})}{\gamma t}$

Results on the NYSE(O) dataset



Results on the NYSE(O) dataset



Conclusions

Conclusions and Future Work

Contributions:

- OGD in online portfolio optimization
- Analysis of the total regret of OGD
- Experimental campaign on real data

Future Works:

- Develop cost aware algorithms (work in progress)
- Generalize the regret analysis to other online learning algorithms

References

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

Cover, T. and Ordentlich, E. (1996). Universal portfolios with side information. *IEEE Transactions on Information Theory*, 42(2):348–363.

Das, P., Johnson, N., and Banerjee, A. (2013). Online lazy updates for portfolio selection with transaction costs. In *AAAI*, pages 202–208.

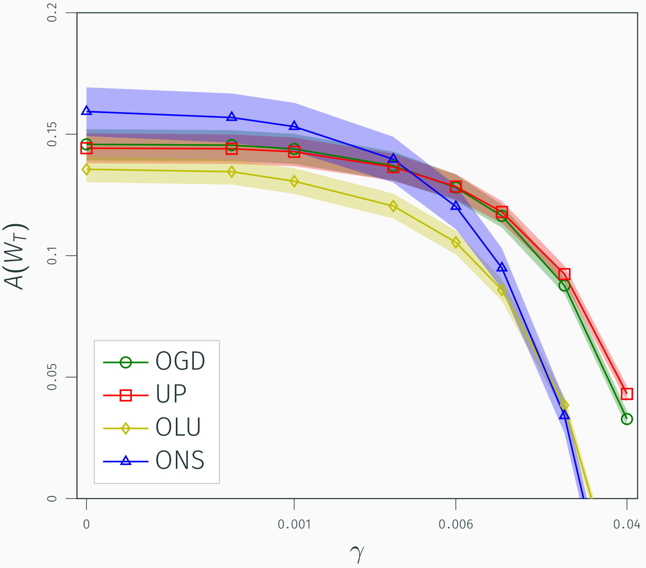
Li, B., Hoi, S., Sahoo, D., and Liu, Z. (2015). Moving average reversion strategy for on-line portfolio selection. *ARTIF INTELL*, 222:104–123.

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

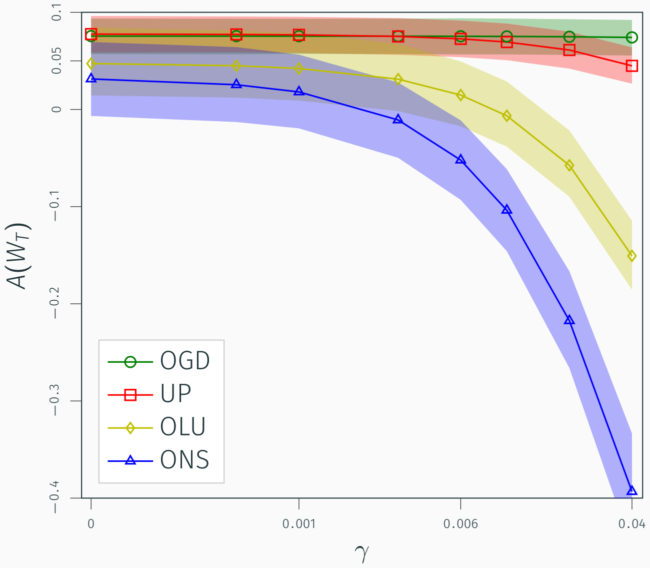
Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936.

Appendix

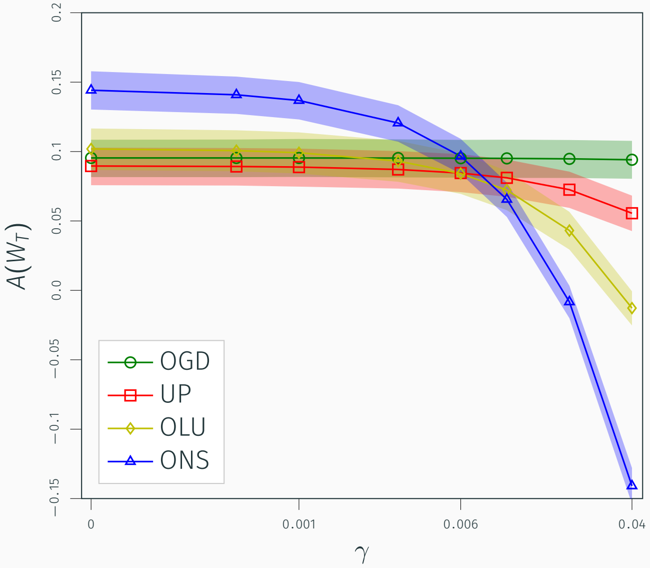
Results on the SP500 dataset (proportional transaction costs)



Results on the TSE dataset



Results on the SP500 dataset



Comparison of Algorithms (single run of NYSE(O))