# Online Gradient Descent for Online Portfolio Optimization with Transaction Costs

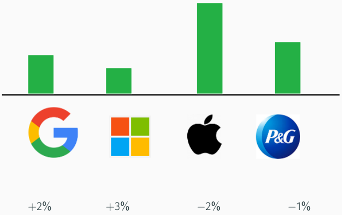
Martino Nicoló Bernasconi de Luca
Supervisor: Francesco Trovó
Co-supervisors: Marcello Restelli, Edoardo Vittori

POLITECNICO
MILANO 1863

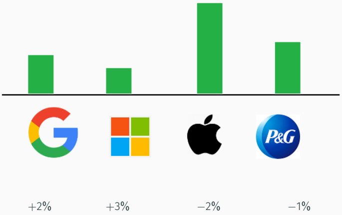
# Table of contents

# Context

# Online Portfolio Optimization



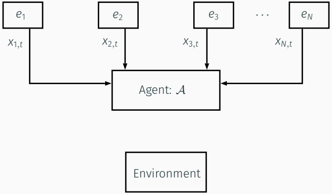
# Online Portfolio Optimization



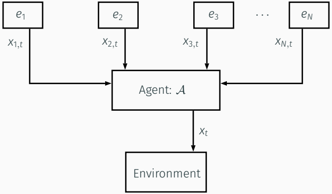

- Modern Portfolio Optimization:
  - Statistical assumptions on the stock dynamics
  - Backward looking
- Online Portfolio Optimization:
  - Adversarial (no assumptions on the distribution)
  - Minimal assumptions on the stock returns

# Context: Online Learning with Experts Advice

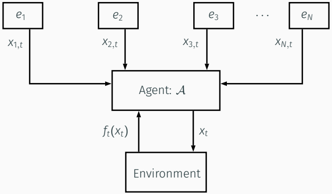
# Context: Online Learning with Experts Advice



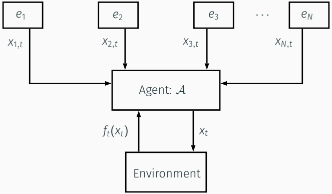
# Context: Online Learning with Experts Advice



# Context: Online Learning with Experts Advice

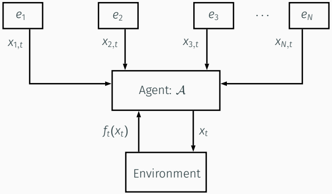


# Context: Online Learning with Experts Advice




- $L_T(\mathcal{A}) := \sum_{t=1}^{T} f_t(x_t)$
- $L_T^* := \inf_{e \in \mathcal{E}} \sum_{t=1}^{T} f_t(x_{e,t})$
- Regret: $R_T = L_T - L_T^*$
- Computational Complexity

# Context: Online Learning with Experts Advice




- $L_T(\mathcal{A}) := \sum_{t=1}^{T} f_t(x_t)$
- $L_T^* := \inf_{e \in \mathcal{E}} \sum_{t=1}^{T} f_t(x_{e,t})$
- Regret: $R_T = L_T - L_T^*$
- Computational Complexity

**No Regret**

$\mathcal{A}$ learns if has per round zero regret asymptotically: $R_T = o(T)$ for any sequence $f_1, f_2, \ldots,$ of losses

# Problem Formulation

# 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
- $\mathcal{E}$ are experts playing $\mathbf{x}_t = \mathbf{x}$ at each turn (Constant Rebalancing Portfolios (CRP))

# 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
- $\mathcal{E}$ are experts playing $\mathbf{x}_t = \mathbf{x}$ at each turn (Constant Rebalancing Portfolios (CRP))

**Regret in Online Portfolio Optimization**

The regret $R_T$ becomes:

$$R_T = \sup_{\mathbf{x} \in \Delta_{N-1}} \log(W_T(\mathbf{x}, \ldots, \mathbf{x}) / W_T(\mathbf{x}_1, \ldots, \mathbf{x}_T)),$$

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 obtained by playing the portfolio sequence $\mathbf{x}_1, \ldots, \mathbf{x}_T$

# Adding Trading Costs

**Regret in Online Portfolio Optimization with Transaction Costs**

$$R_T^C = \underbrace{\sum_{t=1}^{N} f(\mathsf{x}_t, \mathsf{y}_t) - \inf_{\mathsf{x} \in \Delta_{N-1}} \sum_{t=1}^{N} f(\mathsf{x}, \mathsf{y}_t)}_{R_T\text{: standard regret}} + \gamma \underbrace{\sum_{t=1}^{N} ||\mathsf{x}_t - \mathsf{x}_{t-1}||_1}_{C_T\text{: transaction costs}},$$

where $\gamma$ is the transaction rate

$\gamma$ is the proportional transaction rate for buying and selling stocks

# State of the Art

# State of the Art Algorithms

- Algorithms for Online Portfolio Optimization:
  - Universal Portfolios (UP) [Cover and Ordentlich (1996)]
  - Online Newton Step (ONS) [Agarwal et al. (2006)]
- Algorithms for Online Portfolio Optimization with transaction costs:
  - Online Lazy Updates [Das et al. (2013)]

# Proposed Solution

# Online Gradient Descent

**Algorithm 1** OGD in Online Portfolio Optimization with Transaction Costs

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

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

## Standard 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:

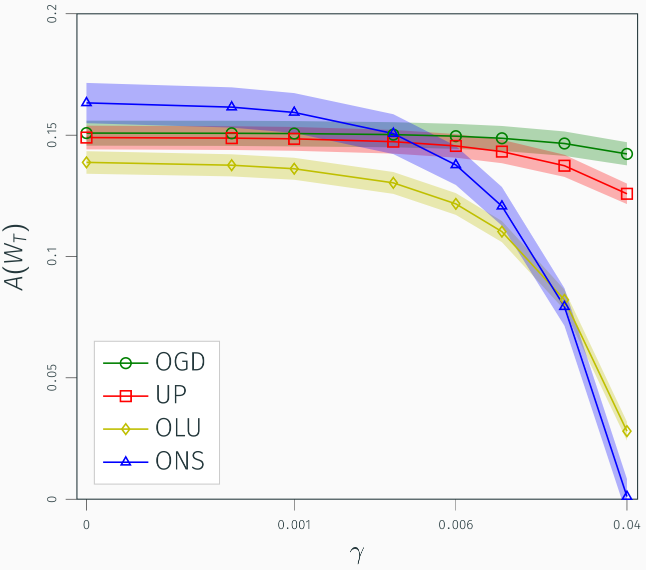
Sample for 100 times, 5 randomly assets, and for different values of the transaction rate parameter $\gamma$

- Average Annual Percentage Yield (APY) defined as

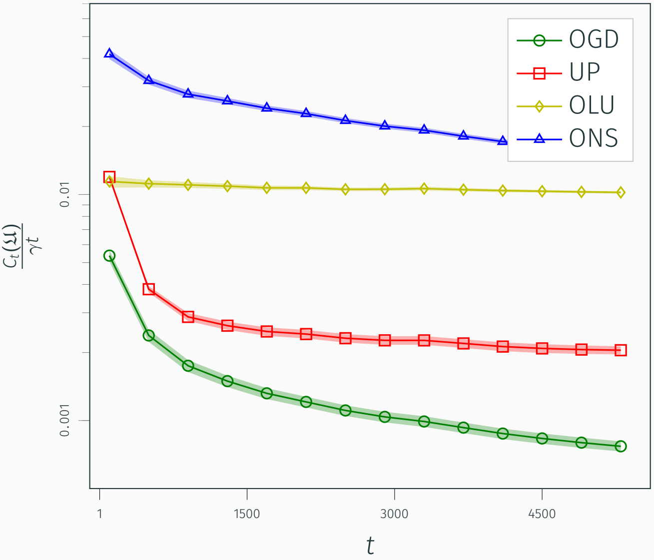
$$A(W_T) = W_T^{250/T} - 1$$

- 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:

- Extended the analysis of to the OGD algorithm to the total regret framework
- Experimental Campaign on Real data

Future Works:

- Develop costs aware algorithms (work in progress)
- Generalize the costs analysis to other algorithms

# References

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

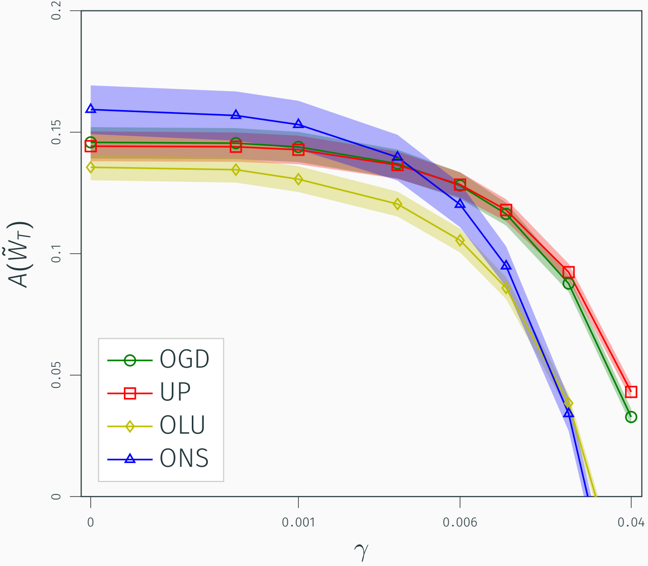
Cover, T. and Ordentlich, E. (1996). Universal portfolios with side information. *IEEE Transactions on Information Theory*, 42(2):348–363.

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

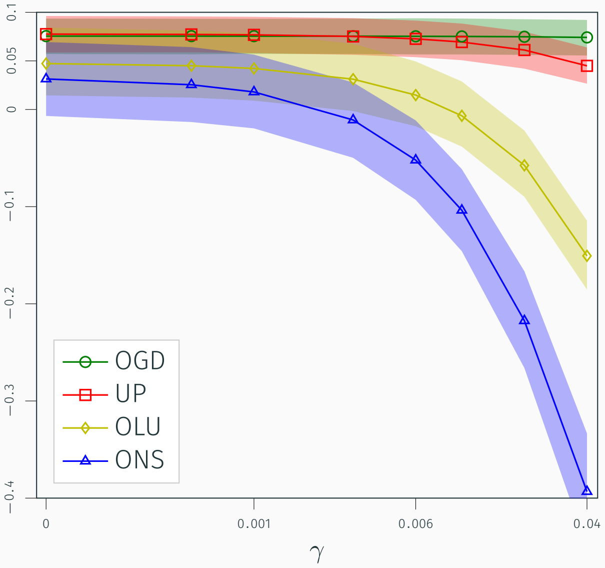
Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936.

# Appendix

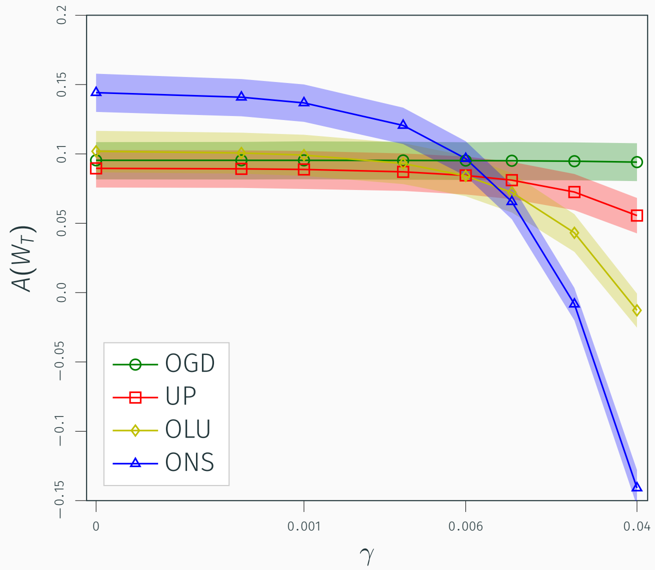
# Results on the SP500 dataset (proportional transaction costs)



# Results on the TSE dataset



# Results on the SP500 dataset



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