Online Lazy Updates for Portfolio Selection with Transaction Costs

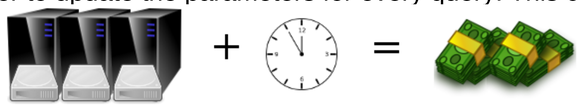
Puja Das, Nicholas Johnson, Arindam Banerjee
Department of Science and Engineering
University of Minnesota
Minneapolis, MN

Section 1: What is the problem?

Problem: Updating billions of model parameters frequently is expensive.

Example:

1. Google brain is an extremely large neural network with millions of nodes and connections. Servers are dedicated to updating the parameters for this neural network. The servers and hard drives have to run constantly in order to update the parameters for every query. This costs a lot of money due to energy consumption.



2. Stock traders who trade many stocks often during a trading period incur a lot of transaction costs.

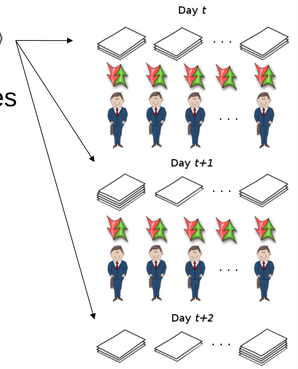
Section 2: Online Portfolio Selection

Choose portfolio based on past stock performance: $\mathbf{p}_t = \langle p_t(1), \ldots, p_t(n) \rangle$

Price relatives $x_t(i)$ are the multiplicative factor by which a stock price changes

- Loss: $x_t(i) < 1$ implies a loss
- Gain: $x_t(i) > 1$ implies a gain
- $x_t(i) = 1$ implies the price remained unchanged

Maximize wealth gain: $S_T(\mathbf{p}_{1:T}, \mathbf{x}_{1:T}) = \prod_{t=1}^{T} (\mathbf{p}_t^T \mathbf{x}_t)$



Section 3: What is the solution?

Solution: Only update certain model parameters infrequently. Sparse updates can approximate optimal solutions for significantly lower costs.

General optimization solution: $\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{t=1}^{T} f_t(\theta)$ where $f_t(\theta) = \phi_t(\theta) + \boxed{\gamma ||\theta - \theta_{t-1}||_1}$

Portfolio Selection solution: $\mathbf{p}_{t+1} = \underset{\mathbf{p} \in \triangle_n}{\operatorname{argmin}} -\eta \log(\mathbf{p}^T \mathbf{x}_t) + \boxed{\alpha ||\mathbf{p} - \mathbf{p}_t||_1} + \frac{1}{2} ||\mathbf{p} - \mathbf{p}_t||_2^2$ Eq. 1

Log gain: maximize logarithmic gain in wealth (i.e. minimize negative log gain in wealth)
L1 penalty: measures fraction of wealth traded. ⇧ penalty = lazy update, ⇩ penalty = frequent update.
L2 penalty: measures distance between portfolios in Euclidean space (e.g. smooths)

Section 4: Online Lazy Updates (OLU) algorithm

ADMM based primal-dual algorithm to solve Eq. 1.

OLU objective function: $\underset{\mathbf{p} \in \triangle_n, \mathbf{p} - \mathbf{p}_t = z}{\operatorname{argmin}} -\eta \log(\mathbf{p}^T \mathbf{x}_t) + \alpha ||z||_1 + \frac{1}{2} ||\mathbf{p} - \mathbf{p}_t||_2^2$

OLU updates:

$$\mathbf{p}^{(k+1)} = \prod\nolimits_{\triangle_n} \left\{ -\frac{\eta \mathbf{x}_t}{(\beta+1)\mathbf{p}_t^T \mathbf{x}_t} + \mathbf{p}_t + \frac{\beta z^{(k)}}{(\beta+1)} - \frac{\beta u^{(k)}}{(\beta+1)} \right\}$$
$$z^{(k+1)} = S_{\alpha/\beta}(\mathbf{p}^{(k+1)} - \mathbf{p}_t + u^{(k)})$$
$$u^{(k+1)} = u^{(k)} + (\mathbf{p}^{(k+1)} - \mathbf{p}_t - z^{(k+1)}) .$$

where $\prod_{\triangle_n}$ is a projection to the simplex and S_ρ is the shrinkage operator.

Advantages of using ADMM:

1. Allows decoupling non-smooth L1 term from smooth terms = Computational benefits (e.g. parallelizable)
2. Fast convergence (in practice)

Section 5: Results

Datasets: NYSE (36 stocks active from 1962-1984) and S&P500 (263 stocks active from 1990-2010).

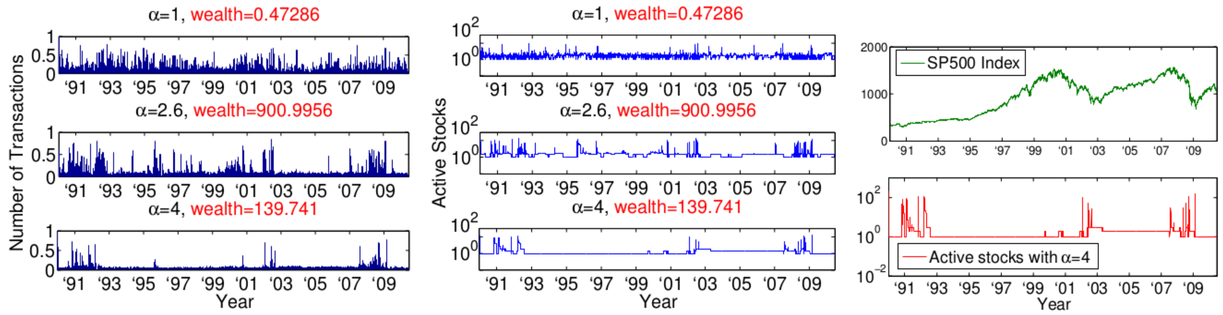


Figure 1: Behavior of OLU as α increases for the number of transactions per day, active stocks, and market movements.

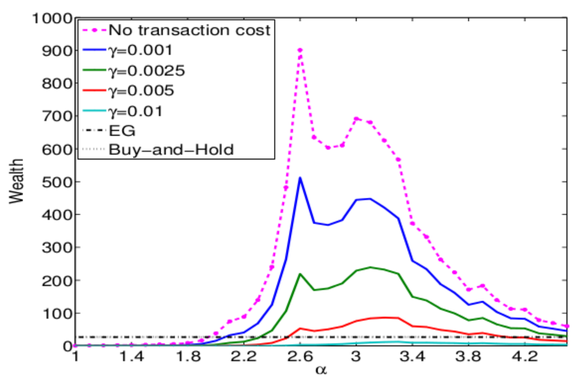


Figure 2: Transactions cost-adjusted wealth.

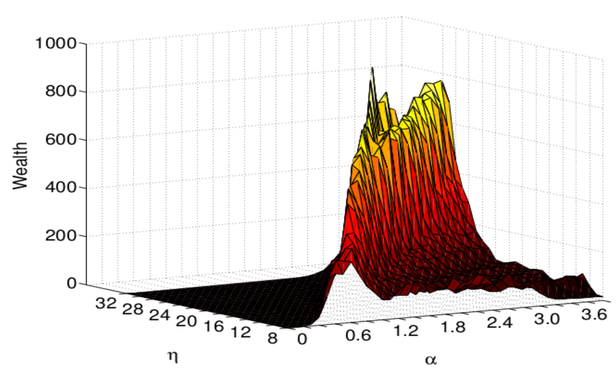


Figure 3: Transaction cost-adjusted wealth as a function of η and α