Structured Hedging for Resource Allocations with Leverage

Nicholas Johnson and Arindam Banerjee

University of Minnesota

ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) 2015.

Motivating Example

- Problem: Given a student salary $, i.e., a small amount, how do we invest in the stock market to make a lot of money?

Motivating Example

- Problem: Given a student salary **$**, i.e., a small amount, how do we invest in the stock market to make a lot of money?
- Solution: Ask for a loan from friends, family, the bank, etc. to increase investment power **$$$**.

Motivating Example

- Problem: Given a student salary **$**, i.e., a small amount, how do we invest in the stock market to make a lot of money?
- Solution: Ask for a loan from friends, family, the bank, etc. to increase investment power **$$$**.
- Some questions:

Motivating Example

- Problem: Given a student salary **$**, i.e., a small amount, how do we invest in the stock market to make a lot of money?
- Solution: Ask for a loan from friends, family, the bank, etc. to increase investment power **$$$**.
- Some questions:
  - How big of a loan?

Motivating Example

- Problem: Given a student salary **$**, i.e., a small amount, how do we invest in the stock market to make a lot of money?
- Solution: Ask for a loan from friends, family, the bank, etc. to increase investment power **$$$**.
- Some questions:
  - How big of a loan?
  - What about risk?

Motivating Example

- Problem: Given a student salary **$**, i.e., a small amount, how do we invest in the stock market to make a lot of money?
- Solution: Ask for a loan from friends, family, the bank, etc. to increase investment power **$$$**.
- Some questions:
  - How big of a loan?
  - What about risk?
  - How can we design an algorithm to do this from data?

Resource Allocation

- Problem: Given n objects the goal is to find a strategy $\mathbf{p}$ which determines how to split up a resource over the n objects.
  - Resources: people, CPUs, money, and products.
  - Objects: software teams, server requests, assets, and warehouses.
- Examples: Job scheduling for compute servers, advertisement and recommendation serving, portfolio selection.
- Such problems often need to be solved online.

Key Aspects

- Leverage
- Long and short positions
- Structured hedging

Leverage

- Resource allocation problems have focused on budgeting resources currently in possession.
- Many problems allow borrowing resources to use as leverage to increase performance.
  - Contractors to help on software projects.
  - Compute servers to handle high demand.
  - Money to increase investment power.

Long and Short Positions

- Many problems have different allocation types or positions.
- For example, in finance we can hold long and short positions.
  - **Long positions**: Purchase shares of stock using cash. Profit if price of shares *increases*.
  - **Short positions**: Borrow shares from bank, sell shares at price X, purchase shares at price Y, and give back shares to bank. Profit if $Y < X$, i.e., price of shares *decreases*.

Hedging

- Long and short positions are opposing positions which we can use to alleviate risk in structurally dependent assets, i.e., hedging.
- Example: hold a long position in s_1 and a short position in s_2. If both crash, there will be a **loss** in the long position s_1 but a **gain** in the short position s_2.
- Question: How to determine which positions to hold and in which stocks?
- Answer: Consider structurally dependent stocks described via a graph.

Structured Hedging

- We consider n assets where the goal is to find an allocation $\mathbf{p} \in \mathcal{P} \subset \mathbb{R}^{2n}$ over the long/short positions and assets such that a convex objective $f(\mathbf{p})$ is minimized.
- Example: With $n = 2$ assets, we describe $\mathbf{p}$ as

$$\begin{matrix} \text{long positions} \left\{ \vphantom{\begin{matrix} a \\ b \end{matrix}} \right. \\ \text{short positions} \left\{ \vphantom{\begin{matrix} a \\ b \end{matrix}} \right. \end{matrix} \underbrace{\begin{bmatrix} q_\ell(1) \\ q_\ell(2) \\ q_s(1) \\ q_s(2) \end{bmatrix}}_{\mathbf{p}} = \underbrace{\begin{bmatrix} q_\ell(1) \\ q_\ell(2) \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{q}_\ell \geq 0} + \underbrace{\begin{bmatrix} 0 \\ 0 \\ q_s(1) \\ q_s(2) \end{bmatrix}}_{\mathbf{q}_s \leq 0}$$

Structured Hedging

- Given graph $\mathcal{G}$ with weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{2n \times 2n}$ and long/short positional vectors $\mathbf{q}_\ell, \mathbf{q}_s \in \mathbb{R}^{2n}$, we introduce a strongly convex hedging penalty function

$$\begin{aligned} \Omega_h &= \sum_{i=1}^{n} \sum_{\substack{j=1+n \\ j \neq i+n}}^{2n} W_{ij} \left(q_\ell(i) + q_s(j) \right)^2 \\ &= \mathbf{p}^\top \mathbf{L} \mathbf{p} \end{aligned}$$

Structured Hedging

- Thus, for the purposes of structured hedging in resource allocation, we will consider problems of the form

$$\min_{\mathbf{p}\in\mathcal{P}} \ell(\mathbf{p}) + \beta\Omega_h(\mathbf{p}) \quad (1)$$

where $\Omega_h(\mathbf{p}) = \mathbf{p}^\top \mathbf{L}\mathbf{p}$.

Online Resource Allocation

- Many resource allocation problems need to be solved dynamically and repeatedly, i.e., online.
- We can pose this as an Online Convex Optimization (OCO) problem with objective function $f_t : \mathcal{P} \to \mathbb{R}$.
- In OCO, optimization proceeds in rounds where at round t the algorithm has to pick a solution from a feasible set $\mathbf{p}_t \in \mathcal{P}$ without knowing $f_t(\cdot)$ and incur a loss of $f_t(\mathbf{p}_t)$.
- The problem is then

$$\min_{\mathbf{p} \in \mathcal{P}} \ell_t(\mathbf{p}) + \beta \Omega_h(\mathbf{p}) \ . \tag{2}$$

Regret

- In order to accomplish a sub-linear regret, we consider solving a linearized version using a first-order Taylor expansion of f_t at $\mathbf{p}_t$ along with a proximal term

$$\mathbf{p}_{t+1} := \underset{\mathbf{p} \in \mathcal{P}}{\operatorname{argmin}} \ \langle \mathbf{p}, \nabla \ell_t(\mathbf{p}_t) \rangle + \beta \Omega_h(\mathbf{p}) + d(\mathbf{p}, \mathbf{p}_t) \ , \tag{3}$$

where $d(\mathbf{p}, \mathbf{p}_t) = \frac{1}{2\eta_t} \|\mathbf{p} - \mathbf{p}_t\|_2^2$.

Online Portfolio Selection

- Consider a stock market consisting of n stocks $\{s(1), \ldots, s(n)\}$ over a span of T days.
- Price relatives: $x_t(i) = \frac{\text{closing price}}{\text{opening price}}$, $\hat{\mathbf{x}}_t = [x_t(1) \ldots x_t(n)]^\top$, and $\mathbf{x}_t = [\hat{\mathbf{x}}_t, \hat{\mathbf{x}}_t]^\top$.
- Portfolio: $\mathbf{p}_t = [p_t(1) \ldots p_t(2n)]^\top \in \mathcal{P}$ where the first $|\ell|$ elements are long-only positions, i.e., $p_t(i) \geq 0$ and the last $|s|$ elements are short-only positions, i.e., $p_t(i) \leq 0$.
- Then $\mathbf{p}$ prescribes investing $p_t(i)$ fraction of the current cash including leverage in stock $s(i)$.

Long and Short Portfolios

- For portfolios that allow both long and short positions with leverage, the multiplicative gain in wealth at the end of the day t is

$$\underbrace{\mathbf{q}_\ell^\top \mathbf{x}_t}_{\text{market change in wealth}} + \underbrace{(1 - \mathbf{q}_\ell^\top \mathbf{1})(1 + r)}_{\text{cash borrowed or not invested}} + \underbrace{\mathbf{q}_s^\top (\mathbf{x}_t - 1)}_{\text{market change in wealth}} + \underbrace{\mathbf{q}_s^\top \mathbf{1} r}_{\text{interest owed on borrowed shares}} .$$

- Logarithmic gain in wealth:

$$LS_T(\mathbf{p}_{1:T}, \mathbf{x}_{1:T}) = \sum_{t=1}^{T} \log\Big(\mathbf{q}_\ell^\top \mathbf{x}_t + \mathbf{q}_s^\top (\mathbf{x}_t - 1 + r) + (1 - \mathbf{q}_\ell^\top \mathbf{1})(1 + r)\Big) .$$

Long and Short Portfolios

- To avoid financial ruin we make an assumption on the price relatives: for $B_\ell, B_s \geq 0$, $0 < 1 - B_\ell < \mathbf{x}_t < 1 + B_s < \infty$.
- We do not assume $B_\ell = B_s$ therefore we define a halfspace that guarantees no-ruin as $\|\mathbf{q}_\ell\|_1 + \frac{B_s + r}{B_\ell + r}\|\mathbf{q}_s\|_1 \leq \frac{1+r}{B_\ell + r} \equiv \mathbf{a}^\top \mathbf{p} \leq \frac{1+r}{B_\ell + r}$.

Structured Hedging Resource Allocations with Leverage (SHERAL)

- Online Portfolio Selection is a special case of (3) with

$$\ell_t(\mathbf{p}_t) = -\log\Big(\alpha_1 \mathbf{q}_\ell^\top \mathbf{x}_t + \alpha_2 \mathbf{q}_s^\top (\mathbf{x}_t - 1 + r) + (1 - \mathbf{q}_\ell^\top \mathbf{1})(1 + r)\Big)$$

  where α_1, α_2 are parameters that control the importance of long and short positions respectively.

- Letting $\eta_t = \eta$ and multiplying each term in (3) by η so that $\lambda = \eta\beta$, then the online portfolio selection with structured hedging problem is

$$\min_{\substack{\mathbf{q}_\ell \geq 0 \\ \mathbf{q}_s \leq 0 \\ \mathbf{a}^\top \mathbf{p} \leq \frac{1+r}{B_\ell + r}}} \eta\langle \mathbf{p}, \nabla \ell_t(\mathbf{p}_t)\rangle + \lambda \mathbf{p}^\top \mathbf{L} \mathbf{p} + \frac{1}{2}\|\mathbf{p} - \mathbf{p}_t\|_2^2 \ . \tag{4}$$

SHERAL

- We propose a projected gradient descent algorithm for solving (4) by computing the gradient, setting it equal to zero, and solving for $\mathbf{p}$

$$\mathbf{p}_{t+1} = \prod_{\mathcal{P}} \left(\eta \nabla \ell_t(\mathbf{p}_t) + \mathbf{p}_t\right) \left(\lambda(\mathbf{L} + \mathbf{L}^\top) + \mathbf{I}\right)^{-1} \tag{5}$$

where

$$\nabla \ell_t(\mathbf{p}_t) = \frac{\alpha_1 \mathbf{D}_\ell^\top \mathbf{x}_t + \alpha_2 \mathbf{D}_s^\top (\mathbf{x}_t - 1 + r) - \mathbf{D}_\ell^\top \mathbf{1}(1+r)}{\alpha_1 \mathbf{p}_t^\top \mathbf{D}_\ell^\top \mathbf{x}_t + \alpha_2 \mathbf{p}_t^\top \mathbf{D}_s^\top (\mathbf{x}_t - 1 + r) + (1 - \mathbf{p}_t^\top \mathbf{D}_\ell^\top \mathbf{1})(1+r)} \tag{6}$$

and $\prod_{\mathcal{P}}$ is a projection onto the convex constraint set.

Experimental Setting

- Initial investment of \$1, invested uniformly with maximum leverage over the positions.
- Daily interest rate $r = 0.000245$ which is equivalent to a yearly interest rate of 6.3%.
- Graph $\mathcal{G}$ is constructed by computing the linear correlation coefficient between all stocks using the previous δ days of $\mathbf{x}_t$.

Exponentiated Gradient (EG)

- One of the first algorithms in this class used for online portfolio selection.
- Solves the problem

$$\mathbf{p}_{t+1} = \underset{\mathbf{p} \in \Delta_n}{\operatorname{argmin}} \ \eta \langle \mathbf{p}, \nabla \ell_t(\mathbf{p}_t) \rangle + d(\mathbf{p}, \mathbf{p}_t)$$

where $d(\mathbf{p}, \mathbf{p}_t) = KL(\mathbf{p} \| \mathbf{p}_t)$ and has solution

$$p_{t+1}(i) \propto p_t(i) \exp(-\eta \nabla \ell_t(i)) \ .$$

Helmbold et al. On-Line Portfolio Selection Using Multiplicative Updates. Mathematical Finance, 1998.

Leveraged Long and Short EG

- Developed various EG* variant portfolios and experimented to observe affect of:

Leveraged Long and Short EG

- Developed various EG* variant portfolios and experimented to observe affect of:
  - Leverage

Leveraged Long and Short EG

- Developed various EG* variant portfolios and experimented to observe affect of:
  - Leverage
  - Long positions

Leveraged Long and Short EG

- Developed various EG* variant portfolios and experimented to observe affect of:
  - Leverage
  - Long positions
  - Short positions

Leveraged Long and Short EG

- Developed various EG* variant portfolios and experimented to observe affect of:
  - Leverage
  - Long positions
  - Short positions
- EG uses KL divergence as the proximal term where $\mathbf{p} \in \Delta_n = \{p(i) \geq 0 \, \forall i, \sum_i p(i) = 1\}$.

Leveraged Long and Short EG

- Developed various EG* variant portfolios and experimented to observe affect of:
  - Leverage
  - Long positions
  - Short positions
- EG uses KL divergence as the proximal term where $\mathbf{p} \in \Delta_n = \{p(i) \geq 0 \, \forall i, \sum_i p(i) = 1\}$.
- To allow $p(i) < 0$, we instead use $d(\mathbf{p}, \mathbf{p}_t) = \frac{1}{2}\|\mathbf{p} - \mathbf{p}_t\|_2^2$ with $\mathbf{p} \in \mathcal{P}$ for the EG* variants.

Datasets

- Dow Jones Industrial Average (DJIA): 30 stocks, 507 trading days between 2001-2003.
- New York Stock Exchange (NYSE): 36 stocks, 5651 trading days between 1962-1984.
- Standard & Poor's 500 ($S\&P500^a$): 25 stocks, 1276 trading days between 1998-2003 (dot-com crash).
- Standard & Poor's 500 ($S\&P500^b$): 263 stocks, 505 trading days between 2007-2009 (financial/housing crash).
- Toronto Stock Exchange (TSX): 88 stocks, 1259 trading days between 1994-1998.

Datasets very different in nature:

- DJIA: 25 out of 30 stocks (83%) lost value.
- NYSE: every stock gained value.
- $S\&P500^a$: 7 out of 25 stocks (28%) lost value.
- $S\&P500^b$: 253 out of 263 stocks (96%) lost value.
- TSX: 32 out of 88 stocks (36%) lost value.

Leveraged Long and Short EG Results

	DJIA	NYSE	SP500[a]	SP500[b]	TSX
EG	0.81	26.70	1.64	0.68	1.59
EG* (LO)	1.55	**6.9×10^{14}**	**20.90**	2.21	**1.0×10^{3}**
EG* (SO)	0.63	0.04	0.34	1.10	1.07
EG* (LS)	**2.00**	6.6×10^{14}	20.65	**2.26**	1.62

Table: Cumulative wealth of EG and EG* variants for each of the five datasets.

EG* (LO) earns $690 trillion!

Comparison with Benchmark Algorithms

- Benchmark algorithms:

Comparison with Benchmark Algorithms

- Benchmark algorithms:
  - Uniform buy-and-hold (U-BAH)

Comparison with Benchmark Algorithms

- Benchmark algorithms:
  - Uniform buy-and-hold (U-BAH)
  - Uniform constant rebalanced portfolio (U-CRP)

Comparison with Benchmark Algorithms

- Benchmark algorithms:
  - Uniform buy-and-hold (U-BAH)
  - Uniform constant rebalanced portfolio (U-CRP)
  - Universal portfolio (UP)

Comparison with Benchmark Algorithms

- Benchmark algorithms:
  - Uniform buy-and-hold (U-BAH)
  - Uniform constant rebalanced portfolio (U-CRP)
  - Universal portfolio (UP)
  - Online lazy updates (OLU)

Comparison with Benchmark Algorithms

- Benchmark algorithms:
  - Uniform buy-and-hold (U-BAH)
  - Uniform constant rebalanced portfolio (U-CRP)
  - Universal portfolio (UP)
  - Online lazy updates (OLU)
  - Best stock

Comparison with Benchmark Algorithms

- Benchmark algorithms:
  - Uniform buy-and-hold (U-BAH)
  - Uniform constant rebalanced portfolio (U-CRP)
  - Universal portfolio (UP)
  - Online lazy updates (OLU)
  - Best stock
- Benchmark algorithm leveraged long and short variants:

Comparison with Benchmark Algorithms

- Benchmark algorithms:
  - Uniform buy-and-hold (U-BAH)
  - Uniform constant rebalanced portfolio (U-CRP)
  - Universal portfolio (UP)
  - Online lazy updates (OLU)
  - Best stock
- Benchmark algorithm leveraged long and short variants:
  - U-BAH LO, SO, LS

Comparison with Benchmark Algorithms

- Benchmark algorithms:
  - Uniform buy-and-hold (U-BAH)
  - Uniform constant rebalanced portfolio (U-CRP)
  - Universal portfolio (UP)
  - Online lazy updates (OLU)
  - Best stock
- Benchmark algorithm leveraged long and short variants:
  - U-BAH LO, SO, LS
  - U-CRP LO, SO, LS

Comparison with Benchmark Algorithms

- Benchmark algorithms:
  - Uniform buy-and-hold (U-BAH)
  - Uniform constant rebalanced portfolio (U-CRP)
  - Universal portfolio (UP)
  - Online lazy updates (OLU)
  - Best stock
- Benchmark algorithm leveraged long and short variants:
  - U-BAH LO, SO, LS
  - U-CRP LO, SO, LS
  - Best stock LO, SO

Comparison with Benchmark Algorithms Results

	DJIA	NYSE	SP500[a]	SP500[b]	TSX
U-BAH	0.76	14.49	1.34	0.63	1.61
U-CRP	0.81	26.78	1.64	0.69	1.59
UP	0.80	26.99	1.62	0.69	1.59
OLU	0.84	50.80	2.45	3.02	2.24
Best Stock (LO)	1.18	54.14	3.77	1.74	6.27
U-BAH* (LO)	0.55	38.44	0.44	0.38	1.71
U-BAH* (SO)	0.43	~ 0	0.01	1.21	0.54
U-BAH* (LS)	0.97	0.43	0.78	1.03	1.12
U-CRP* (LO)	0.61	695.59	1.04	0.43	1.68
U-CRP* (SO)	0.28	~ 0	0.01	0.97	0.54
U-CRP* (LS)	0.83	0.05	0.45	0.93	0.92
Best Stock* (LO)	1.09	65.71	0.56	1.15	7.84
Best Stock* (SO)	1.13	~ 0	0.02	4.71	2.11
SHERAL ($\lambda > 0$)	**2.47**	$\mathbf{1.81 \times 10^{15}}$	**19.89**	**7.84**	**8.74**

Table: Cumulative wealth (without transaction costs) of SHERAL, benchmark algorithms, and several variants for each of the five datasets.

Comparison with Benchmark Algorithms Results

	DJIA	NYSE	SP500[a]	SP500[b]	TSX
U-BAH	0.76	14.49	1.34	0.63	1.61
U-CRP	0.81	26.78	1.64	0.69	1.59
UP	0.80	26.99	1.62	0.69	1.59
OLU	0.84	50.80	2.45	3.02	2.24
Best Stock (LO)	1.18	54.14	3.77	1.74	6.27
U-BAH* (LO)	0.55	38.44	0.44	0.38	1.71
U-BAH* (SO)	0.43	~ 0	0.01	1.21	0.54
U-BAH* (LS)	0.97	0.43	0.78	1.03	1.12
U-CRP* (LO)	0.61	695.59	1.04	0.43	1.68
U-CRP* (SO)	0.28	~ 0	0.01	0.97	0.54
U-CRP* (LS)	0.83	0.05	0.45	0.93	0.92
Best Stock* (LO)	1.09	65.71	0.56	1.15	7.84
Best Stock* (SO)	1.13	~ 0	0.02	4.71	2.11
SHERAL ($\lambda > 0$)	**2.47**	$\mathbf{1.81 \times 10^{15}}$	**19.89**	**7.84**	**8.74**

Table: Cumulative wealth (without transaction costs) of SHERAL, benchmark algorithms, and several variants for each of the five datasets.

SHERAL earns $1.81 quadrillion!

Conclusions

1. Developed an online resource allocation framework to allow both postive and negative weights and leverage.
2. Presented SHERAL algorithm with structured regularizer to induce hedging in the allocation vectors.
3. Demonstrated experimentally that SHERAL is able to significantly outperform current algorithms in terms of wealth earned.

Conclusions

1. Developed an online resource allocation framework to allow both postive and negative weights and leverage.
2. Presented SHERAL algorithm with structured regularizer to induce hedging in the allocation vectors.
3. Demonstrated experimentally that SHERAL is able to significantly outperform current algorithms in terms of wealth earned.

Thank you!

Conclusions

1. Developed an online resource allocation framework to allow both postive and negative weights and leverage.
2. Presented SHERAL algorithm with structured regularizer to induce hedging in the allocation vectors.
3. Demonstrated experimentally that SHERAL is able to significantly outperform current algorithms in terms of wealth earned.

Thank you!

The research was supported in part by NSF grants IIS-1447566, IIS-1422557, CCF-1451986, CNS-1314560, IIS-0953274, IIS-1029711, and by NASA grant NNX12AQ39A.