

Structured Hedging for Resource Allocations with Leverage

Nicholas Johnson
Department of Computer Science and
Engineering
University of Minnesota
njohnson@cs.umn.edu

Arindam Banerjee
Department of Computer Science and
Engineering
University of Minnesota
banerjee@cs.umn.edu

ABSTRACT

Data mining algorithms for computing solutions to online resource allocation (ORA) problems have focused on budgeting resources currently in possession, e.g., investing in the stock market with cash on hand or assigning current employees to projects. In several settings, one can leverage borrowed resources with which tasks can be accomplished more efficiently and cheaply. Additionally, a variety of opposing allocation types or positions may be available with which one can hedge the allocation to alleviate risk from external changes. In this paper, we present a formulation for hedging online resource allocations with leverage and propose an efficient data mining algorithm (SHERAL). We pose the problem as a constrained online convex optimization problem. The key novel components of our formulation are (1) a loss function for general leveraging and opposing allocation positions and (2) a penalty function which hedges between structurally dependent allocation positions to control risk. We instantiate the problem in the context of portfolio selection and evaluate the effectiveness of the formulation through extensive experiments on five datasets in comparison with existing algorithms and several variants.

Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications—*Data mining*

Keywords

Online learning; Structured learning; Finance

1. INTRODUCTION

Online Resource Allocation (ORA) problems are ubiquitous in modern data mining with problems ranging from investment of resources to a variety of assets, such as investing money in the stock market, to modern systems level challenges, such as job scheduling in compute clusters. Increasingly, such problems need to be solved dynamically and

repeatedly in response to external changes, e.g., movements in the stock market, new jobs for compute servers, etc.

Algorithms for ORA problems have focused on budgeting resources that are currently in possession [9, 10, 11, 15, 18]. For example, financial trading algorithms only invest in the stock market using the current cash on hand. However, in several settings there are opportunities to borrow additional resources at cost of a certain interest rate to use as **leverage** to increase returns. For example, a trader can invest with borrowed cash from a bank with the obligation to pay back the loan plus interest at a later date. Many data mining problems can more efficiently and cheaply be solved when leverage is used, however such leveraged allocations are risky since gains and losses are magnified.

Moreover, several ORA problems allow different allocation types or positions. For instance, when investing in the stock market, one can take a **long** position by purchasing shares in a company with cash on hand. Profit is made when the price of the shares *increases*. Additionally, one can take a **short** position by borrowing shares and selling them on the market at a price X and later purchasing them at a price Y and returning them. Profit is made when the price of purchasing them back is less than the price of selling them, $Y < X$, i.e., the price of the shares *decreases*. When such ORA problems allow opposing allocation positions, one can strategically invest in both positions to reduce risk. This is often referred to as **hedging** and can be done by mining and utilizing correlation structures between assets to hold opposing positions in similar assets to withstand market crashes.

In this paper, we present a novel data mining formulation and algorithm for hedging online resource allocations with leverage (SHERAL). The formulation considers opposing long and short positions over assets such as stocks. The key novel components of our formulation are (1) a loss function for general leveraging and structurally dependent allocation positions and (2) a penalty function which encourages hedging between structurally dependent assets to control risk. We pose the problem as a constrained online convex optimization problem and instantiate it in the context of financial data mining where one has to compute a portfolio of stocks to invest in each day.

When starting with the simple portfolio selection algorithm Exponentiated Gradient (EG) [13], we show that adding leveraged long-only positions leads to orders of magnitude improvements in wealth gain, and adding both long and short positions with leverage leads to even more improvements. The results are consistent across multiple datasets,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.

© 2015 ACM. ISBN 978-1-4503-3664-2/15/08 ...\$15.00.

DOI: <http://dx.doi.org/10.1145/2783258.2783378>.

and SHERAL outperforms several existing methods for the portfolio selection problem.

We arrange the rest of the paper as follows. In Section 2 we discuss related work. In Section 3 we introduce our structured hedging formulation. In Section 4 we introduce portfolio selection and describe the position types and leveraging setting. In Section 5 we instantiate the formulation to portfolio selection, and introduce the algorithm. In Section 6 we discuss the experimental results and conclude in Section 7.

2. RELATED WORK

In the context of online portfolio selection, few papers have considered leverage [1, 4, 13] through buying on margin. However, this was simulated after-the-fact and not considered in their problem setting.

One attempt at including leverage and long/short positions in the problem setting was recently presented in [12]. A few preliminary experiments on one dataset were presented to illustrate their formulation worked. However, they did not consider the structural dependencies between positions or utilize such structure to control risk.

Additionally, little work in online portfolio selection has been done to consider risk in the algorithmic setting [6, 14]. However, previous work [1, 3, 4, 5, 7, 8, 13] have shown their algorithms are guaranteed to perform competitively with certain families of adaptive portfolios even in an adversarial market in a costless setting without making any statistical assumptions regarding the movement of the stocks.

We build off of the setting in [12] by introducing a framework with key changes to the loss function and constraint set to allow more flexible long and short portfolios and to consider structured hedging to alleviate risk in addition to thoroughly exploring the affect similar features have on existing algorithms in the literature.

3. HEDGED RESOURCE ALLOCATIONS

In this section, we introduce a framework for structured hedging with leverage, and consider the online resource allocation problem under such structured hedging. In Section 5, we consider the online portfolio selection problem as an instance of such hedged resource allocation.

3.1 Hedging and Leverage

In many resource allocation problems, one may benefit from borrowing additional resources to increase allocation power. For example, in finance, one often buys on margin which is the act of buying shares of stock using cash that is borrowed. This is achieved by putting some amount of cash into a margin account, which acts as collateral c , after which the bank loans additional cash. When one borrows cash, interest is charged on the loan which is often represented as a percentage of the loan size, e.g., a daily interest rate of $r = 0.000245$ which is equivalent to an annual rate of 0.063. In finance, the borrowed cash used to increase potential return is often referred to as **leverage**.

In addition to leverage, there are often opposing allocation types or positions one may take. For instance, one can budget a fraction of the resource to an object, e.g., using cash to buy shares of stock. In finance, this is what is often referred as holding a **long position** in an asset. One will profit in a long position if the value of the allocation *increases*, i.e., the stock price increases.

In contrast, one can choose an opposing type of allocation where one will profit if the value of the allocation *decreases*. In finance, one such type of allocation is called a **short position**, which is when one allocates a borrowed resource. One obtains a short position by placing collateral c , such as cash, into a margin account after which a resource, such as shares of a stock, is borrowed with the obligation to return it later at its current value. Often interest is charged on the borrowed resource r_b and may be earned on the collateral r_c , however in this paper we assume the interest on the borrowed resource, collateral, and cash leverage are the same, i.e., $r_b = r_c = r$. We also assume that the value of the collateral c is equal to the value of the resource borrowed.

More specifically, a trader places X amount of cash as collateral into a margin account and then the bank borrows the trader shares of stock worth X . The trader then sells the shares at the current market price (presumably X). After the market moves, the trader purchases the shares back at the new market price Y and returns them to the bank. The trader profits if the purchase price is less than the selling price, i.e., $Y < X$.

Long and short positions act in opposition and are most frequently utilized to offset the risk of a particular allocation. For example, if two stocks s_1 and s_2 are positively correlated but highly volatile, then a trader may hold a long position in s_1 and a short position in s_2 . The trader is exposed to less risk with this allocation because if both stocks crash then the trader will not lose as much since there will be a loss in the long position s_1 but a gain in the short position s_2 . This is often referred to as **hedging** and the difficulty is in taking advantage of the structural dependencies between assets to balance high returns and exposure to risk.

3.2 Structured Hedging

We consider structure in resource allocation problems in the form of a graph over the assets. For n objects, the goal is to find an allocation $\mathbf{p} \in \mathcal{P} \subset \mathbb{R}^{2n}$ which determines how to split up a resource amongst long and short positions over the n objects such that a certain (convex) objective $f(\mathbf{p})$ is minimized. We denote the set of indices $\ell = \{1, \dots, n\}$ and $s = \{n+1, \dots, 2n\}$ as the long and short positions in \mathbf{p} and let \mathbf{D}_ℓ and \mathbf{D}_s be $2n \times 2n$ diagonal matrices with $D_\ell(i, i) = 1$ for $i \in \ell$ and $D_s(i, i) = 1$ for $i \in s$ and 0 otherwise. Let $\mathbf{q}_\ell = \mathbf{D}_\ell \mathbf{p} \geq 0$ and $\mathbf{q}_s = \mathbf{D}_s \mathbf{p} \leq 0$ where the inequalities are taken element-wise. \mathbf{q}_ℓ and \mathbf{q}_s are the long-only and short-only vectors of size $2n \times 1$ with value equal to \mathbf{p} for indices in ℓ and s respectively and 0 otherwise.

For example, in the context of portfolio selection, the n objects can be different assets, such as stocks, and \mathbf{p} is an investment strategy, i.e., what fraction of one's total resources (say, money) should one put on each asset (stock). The basic idea of hedging is to place resources in opposing positions and different assets. A simple way to accomplish this is to select a set of assets to hold long positions and another set of assets to hold short positions. A potential issue with such an approach is that there may be structural dependencies between the assets, such as being negatively correlated, which may result in losing every position held.

For example, Apple and Costco may be negatively correlated because Apple sells luxury items and Costco sells consumer staples. Since companies that sell luxury items are cyclical, i.e., share price is positively correlated with economic conditions, the purchases of luxury items often

slows during market crashes. In contrast, companies that sell consumer staples are non-cyclical and the purchases of consumer staple items does not slow during market crashes. Therefore, if we hold a long position in Apple and a short position in Costco during this time, we will lose in both positions.

Such structure is often hard to determine a priori and represent simply such as in groups of assets, i.e., market sectors. We can more easily capture relationships between assets via a graph where the value of the edges determines how similar the assets are. Similarity can be any suitable measure of correlation such as linear correlation, Rank correlation, etc.

The goal of structured hedging is to develop hedged strategies which explicitly consider such graph structured assets.

3.2.1 Hedging Penalty Function

For the development, we assume knowledge of the structured graph $\mathcal{G} = (V, E)$ where $V = \{v(1), \dots, v(n)\}$ are the nodes and $E = \{e(1), \dots, e(m)\}$ are the edges where $e(k) = (v(i), v(j))$ if there is an edge between nodes i and j . Let the weight on edge $e(k)$ be $W_{ij} \in \mathbb{R}$, then $\mathbf{W} \in \mathbb{R}^{n \times n}$ is the graph's weighted adjacency matrix with $W_{ij} = 0$ if there is no edge between nodes i and j . We outline approaches for constructing such a graph directly from the data in Section 5 in the context of portfolio selection.

Given such a graph and long/short position vectors $\mathbf{q}_\ell, \mathbf{q}_s \in \mathbb{R}^{2n}$ where the long position of asset i is contained in index i of \mathbf{q}_ℓ and the short position of asset i is contained in the index $i + n$ of \mathbf{q}_s , we introduce a hedging penalty function

$$\Omega_h = \sum_{i=1}^n \sum_{\substack{j=1+n \\ j \neq i+n}}^{2n} W_{ij} (q_\ell(i) + q_s(j))^2 \quad (1)$$

where W_{ij} is a measure of the similarity between assets i and j and $q_\ell(i) \geq 0$ and $q_s(j) \leq 0$ are the value of the asset positions. We seek to minimize Ω_h , so when W_{ij} is large, we minimize Ω_h by making the value of the assets' opposing positions close, effectively encouraging hedging. When W_{ij} is small, the assets are not similar so the assets' position values do not need to be close and we do not hedge.

One key aspect of hedging is that we want to hold opposite positions in different but structurally related assets rather than the same asset since this is similar to holding only the difference between the positions. (1) is designed to do this by (i) taking into account the structural dependencies between assets in \mathcal{G} and (ii) only considering edges between opposing positions in different assets. In other words, we do not consider the difference between $q_\ell(i)$ and $q_\ell(j)$ (same position type, different assets) or the difference between $q_\ell(i)$ and $q_s(i)$ (different position type, same assets).

One benefit of our quadratic hedging function (1) is that we are able to capture the interaction between different asset *positions* which provides more flexibility in responding to asset fluctuations in addition to considering just the correlation between assets. We can consider other hedging penalty functions, e.g., $\sum_{i=1}^n \sum_{\substack{j=1+n \\ j \neq i+n}}^{2n} W_{ij} \left(\frac{1}{q_\ell(i) - q_s(j)} \right)$, however such a function does not capture the position interactions. Additionally, such functions may not reflect a true hedging strategy since the function can be minimized by making one position large while leaving the other small. One limitation of (1) is that when there are two identical as-

sets, this form will encourage both long positions and short positions to be similar across assets.

We can represent (1) in a more compact form by considering a graph over the $2n$ positions instead of the n assets. Let $\mathcal{G}_p = (V_p, E_p)$ be a graph over the positions where $V_p = \{v_\ell(1), v_s(1), \dots, v_\ell(n), v_s(n)\}$ are the nodes, i.e., $v_\ell(i)$ is the node for the long position of asset i and $v_s(i)$ is the node for the short position of asset i . We can construct the set of edges E_p and their corresponding weights U_{ij} from \mathcal{G} by considering assets i and j that are connected by edge $e(k)$ with weight W_{ij} . We add an edge to E_p between nodes $v_\ell(i)$ and $v_s(j)$ with weight W_{ij} and an edge between nodes $v_s(i)$ and $v_\ell(j)$ with weight W_{ij} . In other words, we only connect the opposing position nodes between the different assets and the corresponding edges will have the same weight as the weight between the assets.

Under this construction we can similarly define \mathcal{G}_p 's weighted adjacency matrix $\mathbf{U} = \begin{bmatrix} \mathbf{0} & \mathbf{W} \\ \mathbf{W} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{2n \times 2n}$, which is symmetric since \mathbf{W} is symmetric. Let $\mathbf{D} \in \mathbb{R}^{2n \times 2n}$ be a diagonal matrix with $D(i, i) = \sum_j U_{ij}$ and 0 otherwise, then

$$\mathbf{L} = \mathbf{D} + \mathbf{U} \quad (2)$$

which is also symmetric and positive semi-definite by design. Then, by construction

$$\mathbf{p}^\top \mathbf{L} \mathbf{p} = \sum_{i=1}^n \sum_{\substack{j=1+n \\ j \neq i+n}}^{2n} W_{ij} (q_\ell(i) + q_s(j))^2. \quad (3)$$

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

$$\min_{\mathbf{p} \in \mathcal{P}} \ell(\mathbf{p}) + \beta \Omega_h(\mathbf{p}) \quad (4)$$

where $\Omega_h(\mathbf{p}) = \mathbf{p}^\top \mathbf{L} \mathbf{p}$ and $\beta \geq 0$.

3.3 Online Resource Allocation

Several resource allocation problems need to be solved online, i.e., dynamically over time. Such a problem can be modeled as an online optimization problem with objective function $f_t(\mathbf{p})$ at time t . In such an online setting, the optimization proceeds in rounds where in 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 $f_t(\mathbf{p}_t)$. Thus, the problem at time t takes the form

$$\min_{\mathbf{p} \in \mathcal{P}} \ell_t(\mathbf{p}) + \beta \Omega_h(\mathbf{p}) \quad (5)$$

where ℓ_t denotes a suitable resource allocation loss. Ideally, we would like to minimize the constrained cumulative loss

$$\sum_{t=1}^T \ell_t(\mathbf{p}_t) + \beta \Omega_h(\mathbf{p}_t). \quad (6)$$

However, in the online setting, minimization of (6) is not feasible since we do not know the sequence of ℓ_t a priori. Alternatively, over T rounds we intend to get a sequence of \mathbf{p}_t such that the following *regret* is sub-linear in T

$$R_T = \sum_{t=1}^T f_t(\mathbf{p}_t) - \min_{\mathbf{p}^*} \sum_{t=1}^T f_t(\mathbf{p}^*) \leq o(T) \quad (7)$$

where $f_t(\mathbf{p}) = \ell_t(\mathbf{p}) + \beta \Omega_h(\mathbf{p})$. The regret is measured *w.r.t* the best fixed minimizer in hindsight \mathbf{p}^* .

Following recent advances in online convex optimization, in order to accomplish a sub-linear regret, in each step $(t+1)$, we consider solving a linearized version of (5) obtained by a first-order Taylor expansion of ℓ_t at \mathbf{p}_t along with a proximal term, so that

$$\mathbf{p}_{t+1} = \underset{\mathbf{p} \in \mathcal{P}}{\operatorname{argmin}} \langle \mathbf{p}, \nabla \ell_t(\mathbf{p}_t) \rangle + \beta \Omega_h(\mathbf{p}) + d(\mathbf{p}, \mathbf{p}_t), \quad (8)$$

where $d(\mathbf{p}, \mathbf{p}_t) = \frac{1}{2\eta_t} \|\mathbf{p} - \mathbf{p}_t\|_2^2$ and parameters $\eta_t, \beta \geq 0$.

4. ONLINE PORTFOLIO SELECTION

We consider a stock market consisting of n stocks $\{s_1, \dots, s_n\}$ over a span of T periods. For ease of exposition, we will consider a period to be a day, but the analysis presented holds for any valid definition of a ‘period’ such as an hour or a month. Let $x_t(i)$ denote the *price relative* of stock s_i in day t , i.e., the multiplicative factor by which the price of s_i changes in day t . Hence, $x_t(i) > 1$ implies a gain, $x_t(i) < 1$ implies a loss, and $x_t(i) = 1$ implies the price remained unchanged. We assume, $x_t(i) > 0 \forall i, t$. Let $\hat{\mathbf{x}}_t = [x_t(1), \dots, x_t(n)]^\top$ denote the vector of price relatives for day t , let $\mathbf{x}_t = [\hat{\mathbf{x}}_t, \hat{\mathbf{x}}_t]^\top$ be the double stacked vector of price relatives, and let $\mathbf{x}_{1:t}$ denote the collection of such price relative vectors up to and including day t . A portfolio on day t is $\mathbf{p}_t = [p_t(1), \dots, p_t(2n)]^\top \in \mathcal{P}$ where the first $|l|$ elements are long-only positions, i.e., $p_t(i) \geq 0$ and the last $|s|$ elements are short-only positions, i.e., $p_t(i) \leq 0$ which prescribes investing $p_t(i)$ fraction of the total wealth, including leverage, in stock s_i . Note that the portfolio \mathbf{p}_t has to be decided before knowing \mathbf{x}_t which will be revealed only at the end of the day.

4.1 Long-Only Portfolios

For long-only portfolios $\mathbf{q}_\ell \geq 0$ without leverage, the multiplicative gain in wealth at the end of day t is $\mathbf{q}_\ell^\top \mathbf{x}_t$. When we allow borrowing cash from the bank as leverage we have

$$\underbrace{\mathbf{q}_\ell^\top \mathbf{x}_t}_{\text{market change in wealth}} + \underbrace{(1 - \mathbf{q}_\ell^\top \mathbf{1})(1+r)}_{\text{cash borrowed or not invested}}. \quad (9)$$

The last term accounts for the percentage of our current wealth we borrowed from the bank or wealth not invested. If we did not borrow any cash then $0 \leq (1 - \mathbf{q}_\ell^\top \mathbf{1}) \leq 1$ and any cash not invested is considered as being held in a savings bank account which earns interest at of rate of r . If we did borrow cash, then this term is negative and is the amount we have to pay back to the bank plus interest r .

When we allow borrowing cash, we have to be careful about owing more money than we have left at the end of any day in order to avoid financial ruin. For instance, if we invest long with leverage and the market crashes, we may have no money left to pay back the bank loan. In order to guarantee no-ruin, we make an assumption on the price relatives similar to [12] such that $0 < 1 - B_\ell < \mathbf{x}_t$ where B_ℓ is a parameter that can be set based on historical stock data. Then the maximum amount we can invest is $\frac{1+r}{B_\ell+r}$. We prove that this will not lead to negative growth rate in the following proposition.

Proposition 1. For a long-only portfolio \mathbf{q}_ℓ such that $\mathbf{q}_\ell \geq 0$, $\|\mathbf{q}_\ell\|_1 \leq \frac{1+r}{B_\ell+r}$, and with bounded price relatives $0 < 1 - B_\ell < \mathbf{x}_t$, the multiplicative gain $\mathbf{q}_\ell^\top \mathbf{x}_t + (1 - \mathbf{q}_\ell^\top \mathbf{1})(1+r) \geq 0$.

PROOF. For ease of exposition, let us consider only a single investment so $\mathbf{q}_\ell \in \mathbb{R}_+$. Then in the worst case we have

$$\begin{aligned} & \mathbf{q}_\ell^\top \mathbf{x}_t + (1 - \mathbf{q}_\ell^\top \mathbf{1})(1+r) \\ & \geq \frac{1+r}{B_\ell+r} (1 - B_\ell) + \left(1 - \frac{1+r}{B_\ell+r}\right) (1+r) \\ & = 0. \end{aligned}$$

□

4.2 Short-Only Portfolios

For short-only portfolios $\mathbf{q}_s \leq 0$ without cash leverage, the multiplicative gain in wealth at the end of day t consists of the difference between the price value of shares borrowed $\mathbf{q}_s^\top \mathbf{1}$ and the price the shares are now worth $\mathbf{q}_s^\top \mathbf{x}_t$ minus the interest owed on borrowing the shares plus the collateral c and interest on collateral r . In this paper, we assume the interest rate for borrowing cash/shares is the same as the interest rate earned for keeping cash and collateral in a margin account. Additionally, the total value of shares borrowed is equal to the collateral put down, i.e., $\|\mathbf{q}_s\|_1 = c$. Therefore, the multiplicative gain in wealth is $\mathbf{q}_s^\top (\mathbf{x}_t - 1) + \mathbf{q}_s^\top \mathbf{1}r - \mathbf{q}_s^\top \mathbf{1}(1+r)$. When we allow borrowing cash from the bank as leverage we have

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

Re-arranging the terms we get

$$\begin{aligned} & \mathbf{q}_s^\top (\mathbf{x}_t - 1) + \mathbf{q}_s^\top \mathbf{1}r + (1 + \mathbf{q}_s^\top \mathbf{1})(1+r) - \mathbf{q}_s^\top \mathbf{1}(1+r) \\ & = \mathbf{q}_s^\top (\mathbf{x}_t - 1 + r) + (1+r). \end{aligned} \quad (10)$$

Similar to long-only portfolios, in order to guarantee no-ruin we assume $\mathbf{x}_t < 1 + B_s < \infty$. Then the maximum amount we can invest is $\frac{1+r}{B_s+r}$. We prove that this will not lead to negative growth rate in the following proposition.

Proposition 2. For a short-only portfolio \mathbf{q}_s such that $\mathbf{q}_s \leq 0$, $\|\mathbf{q}_s\|_1 \leq \frac{1+r}{B_s+r}$, and with bounded price relatives $\mathbf{x}_t < 1 + B_s < \infty$, the multiplicative gain $\mathbf{q}_s^\top (\mathbf{x}_t - 1 + r) + (1+r) \geq 0$.

PROOF. For ease of exposition, let us consider only a single investment so $\mathbf{q}_s \in \mathbb{R}_-$. Then in the worst case we have

$$\begin{aligned} & \mathbf{q}_s^\top (\mathbf{x}_t - 1 + r) + (1+r) \\ & \geq -\frac{1+r}{B_s+r} ((1 + B_s) - 1 + r) + (1+r) \\ & = 0. \end{aligned}$$

□

4.3 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 will combine (9) and (10) to get

$$\mathbf{q}_\ell^\top \mathbf{x}_t + (1 - \mathbf{q}_\ell^\top \mathbf{1})(1+r) + \mathbf{q}_s^\top (\mathbf{x}_t - 1 + r) + (1+r).$$

However, we have counted the amount of cash borrowed or not invested plus interest twice since both were included individually in (9) and (10) so we subtract $(1+r)$ to get

$$\mathbf{q}_\ell^\top \mathbf{x}_t + \mathbf{q}_s^\top (\mathbf{x}_t - 1 + r) + (1 - \mathbf{q}_\ell^\top \mathbf{1})(1+r). \quad (11)$$

Unlike in [12], we do not assume $B_\ell = B_s$ since bear and bull markets tend to not be symmetrical. Therefore, we

Algorithm 1 SHERAL Algorithm

- 1: Input $\mathbf{p}_t, \mathbf{x}_t, \eta, \alpha_1, \alpha_2, \lambda, B_\ell, B_s, r, \mathbf{D}_\ell, \mathbf{D}_s, \mathcal{G}$.
- 2: Compute weighted adjacency matrix \mathbf{L} via (2).
- 3: Compute portfolio for day $t + 1$:

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

where $\nabla \ell_t(\mathbf{p}_t)$ is as defined in (14) and $\prod_{\mathbf{p} \in \mathcal{P}}$ is a projection to the convex set $\mathcal{P} = \left\{ \mathbf{p} \mid \mathbf{D}_\ell \mathbf{p} \geq 0, \mathbf{D}_s \mathbf{p} \leq 0, \mathbf{a}^\top \mathbf{p} \leq \frac{1+r}{B_\ell+r} \right\}$ via alternating projections.

define a hyperplane that guarantees no-ruin for a portfolio with a combination of long and short positions as $\|\mathbf{q}_\ell\|_1 + \frac{B_s+r}{B_\ell+r} \|\mathbf{q}_s\|_1 \leq \frac{1+r}{B_\ell+r}$. Since $\mathbf{q}_s \leq 0$, we can define a vector \mathbf{a} such that the first $|\ell|$ elements are equal to 1 and the last $|s|$ elements are equal to $-\frac{B_s+r}{B_\ell+r}$, so the constraint on maximum investment as a combination of long and short positions is $\mathbf{a}^\top \mathbf{p} \leq \frac{1+r}{B_\ell+r}$. We prove that this will not lead to negative growth rate in the following proposition.

Proposition 3. For a long and short portfolio $\mathbf{p} = \mathbf{q}_\ell + \mathbf{q}_s$ such that $\mathbf{q}_\ell \geq 0, \mathbf{q}_s \leq 0, \mathbf{a}^\top \mathbf{p} \leq \frac{1+r}{B_\ell+r}$, and with bounded price relatives $0 < 1 - B_\ell < \mathbf{x}_t < 1 + B_s < \infty$, the multiplicative gain $\mathbf{q}_\ell^\top \mathbf{x}_t + \mathbf{q}_s^\top (\mathbf{x}_t - 1 + r) + (1 - \mathbf{q}_\ell^\top \mathbf{1})(1 + r) \geq 0$.

PROOF. Setting $u = \|\mathbf{q}_\ell\|_1$ implies $\|\mathbf{q}_s\|_1 \leq \left(\frac{1+r}{B_s+r} - \frac{B_\ell+r}{B_s+r} u \right)$. Then in the worst case we have

$$\begin{aligned} & \mathbf{q}_\ell^\top \mathbf{x}_t + \mathbf{q}_s^\top (\mathbf{x}_t - 1 + r) + (1 - \mathbf{q}_\ell^\top \mathbf{1})(1 + r) \\ & \geq u(1 - B_\ell) - \left(\frac{1+r}{B_s+r} - \frac{B_\ell+r}{B_s+r} u \right) ((1 + B_s) - 1 + r) + (1 - u)(1 + r) \\ & = 0. \end{aligned}$$

□

Therefore, the multiplicative gain in wealth at the end of day t for a leveraged portfolio with long and short positions is (11) which, in the worst case, is non-negative. This ensures we have enough money left over after the market moves to pay back all loans.

Given this, and a sequence of price relatives $\mathbf{x}_{1:t-1} = \{\mathbf{x}_1, \dots, \mathbf{x}_{t-1}\}$ up to day $(t - 1)$, the sequential portfolio selection problem in day t is to determine a portfolio \mathbf{p}_t based on past performance of the stocks. At the end of day t , \mathbf{x}_t is revealed and the actual performance of \mathbf{p}_t gets determined by (11). Over T periods, for a sequence of portfolios $\mathbf{p}_{1:T} = \{\mathbf{p}_1, \dots, \mathbf{p}_T\}$, the multiplicative and logarithmic gain in wealth are

$$\begin{aligned} S_T(\mathbf{p}_{1:T}, \mathbf{x}_{1:T}) &= \prod_{t=1}^T \left(\mathbf{q}_\ell^\top \mathbf{x}_t + \mathbf{q}_s^\top (\mathbf{x}_t - 1 + r) + (1 - \mathbf{q}_\ell^\top \mathbf{1})(1 + r) \right) \\ LST(\mathbf{p}_{1:T}, \mathbf{x}_{1:T}) &= \sum_{t=1}^T \log \left(\mathbf{q}_\ell^\top \mathbf{x}_t + \mathbf{q}_s^\top (\mathbf{x}_t - 1 + r) + (1 - \mathbf{q}_\ell^\top \mathbf{1})(1 + r) \right). \end{aligned}$$

Ideally, for a costless environment (no transaction costs) we would like to maximize $LST(\mathbf{p}_{1:T}, \mathbf{x}_{1:T})$ over $\mathbf{p}_{1:T}$. However, \mathbf{x}_t is not available when one has to decide on \mathbf{p}_t . Therefore, we will use $LST(\mathbf{p}_t, \mathbf{x}_t)$ as the loss function ℓ_t in (8).

Algorithm 2 Hedging for Online Portfolio Selection

- 1: Input $\eta, \alpha_1, \alpha_2, \lambda, B_\ell, B_s$, Interest rate r , Transaction cost γ , Days lag δ .
 - 2: Set $S_0^\gamma = \$1$ and
 - 3: $p_{0:1}(i) = \frac{(1+r)/(B_\ell+r)}{1+(B_s+r)/(B_\ell+r)} \forall i$ (uniform over positions).
 - 4: For $t = 1, \dots, T$
 - 5: Receive the vector of price relatives: \mathbf{x}_t .
 - 6: Compute multiplicative gain in wealth via (11) as Ψ_t .
 - 7: Compute wealth: $S_t^\gamma = S_{t-1}^\gamma (\Psi_t - \gamma \|\mathbf{p}_t - \mathbf{p}_{t-1}\|_1)$.
 - 8: If $t \leq \delta$
 - 9: $\mathbf{p}_{t+1} = \mathbf{p}_t$ (uniform over positions).
 - 10: Else
 - 11: Compute graph: \mathcal{G} .
 - 12: $\mathbf{p}_{t+1} = \text{SHERAL}(\mathbf{p}_t, \mathbf{x}_t, \eta, \alpha_1, \alpha_2, \lambda, B_\ell, B_s, r, \mathbf{D}_\ell, \mathbf{D}_s, \mathcal{G})$.
 - 13: end for
-

5. HEDGING PORTFOLIO SELECTION

Online portfolio selection can now be viewed as a special case of our online resource allocation with structured hedging setting with

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

where $\alpha_1, \alpha_2 \geq 0$ are parameters that control the importance of long and short positions respectively. Note, if both long and short positions are valued equally, $\alpha_2 > \alpha_1$ since the scale of the position returns differ by a factor (Refer to 6.4 (c) for an example). Letting $\eta_t = \eta$ and multiplying each term in (8) by η so that $\lambda = \eta\beta$, the online portfolio selection with structured hedging problem is

$$\min_{\substack{\mathbf{q}_\ell \geq 0 \\ \mathbf{q}_s \leq 0 \\ \mathbf{a}^\top \mathbf{p} \leq \frac{1+r}{B_\ell+r}}} \eta \langle \mathbf{p}, \nabla \ell_t(\mathbf{p}_t) \rangle + \lambda \mathbf{p}^\top \mathbf{L} \mathbf{p} + \frac{1}{2} \|\mathbf{p} - \mathbf{p}_t\|_2^2. \quad (12)$$

This is a strongly convex optimization problem with the linear constraint set $\mathcal{P} = \left\{ \mathbf{p} \mid \mathbf{q}_\ell \geq 0, \mathbf{q}_s \leq 0, \mathbf{a}^\top \mathbf{p} \leq \frac{1+r}{B_\ell+r} \right\}$ where $\mathbf{q}_\ell = \mathbf{D}_\ell \mathbf{p}$ and $\mathbf{q}_s = \mathbf{D}_s \mathbf{p}$ (Section 3.2).

We propose an efficient projected gradient descent algorithm for solving (12). Since the objective in (12) is strongly convex we can find the minimum by taking the gradient, setting it to zero, and solving for \mathbf{p} to get

$$\mathbf{p}_{t+1} = \prod_{\mathcal{P}} (\eta \nabla \ell_t(\mathbf{p}_t) + \mathbf{p}_t) \left(\lambda (\mathbf{L} + \mathbf{L}^\top) + \mathbf{I} \right)^{-1} \quad (13)$$

where

$$\nabla \ell_t(\mathbf{p}_t) = \frac{\alpha_1 \mathbf{D}_\ell^\top \mathbf{x}_t + \alpha_2 \mathbf{D}_s^\top (\mathbf{x}_t - 1 + r) - \mathbf{D}_\ell^\top \mathbf{1}(1 + r)}{\alpha_1 \mathbf{p}_t^\top \mathbf{D}_\ell^\top \mathbf{x}_t + \alpha_2 \mathbf{p}_t^\top \mathbf{D}_s^\top (\mathbf{x}_t - 1 + r) + (1 - \mathbf{p}_t^\top \mathbf{D}_\ell^\top \mathbf{1})(1 + r)} \quad (14)$$

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

Algorithm 1 shows the complete details for computing the hedged portfolio. Algorithm 2 is our hedged online portfolio selection with leverage algorithm which includes computing the transaction-cost adjusted wealth using a fixed percentage transaction cost γ .

5.1 Regret Bound

We sequentially invest with the hedged portfolios $\mathbf{p}_1, \dots, \mathbf{p}_T$ obtained from Algorithm 2 and on day t suffer a loss of $f_t(\mathbf{p}_t) = \eta \ell_t + \lambda \mathbf{p}_t^\top \mathbf{L} \mathbf{p}_t$. Our goal is to minimize the *regret*

with respect to the best fixed portfolio \mathbf{p}^* in hindsight. We establish the standard regret bound in portfolio selection literature [1, 4, 13].

REGRET BOUND 1. *Let $\mathbf{p}^* \in \mathcal{P}$ be the fixed portfolio obtained from $\min_{\mathbf{p}} \sum_{t=1}^T \ell_t(\mathbf{p})$. For $\eta = \sqrt{T}$ and $\|\nabla \ell_t(\mathbf{p}_t)\| \leq G$, the regret can be bounded as,*

$$\eta \sum_{t=1}^T \ell_t(\mathbf{p}_t) + \lambda \mathbf{p}_t^\top \mathbf{L} \mathbf{p}_t - \eta \sum_{t=1}^T \ell_t(\mathbf{p}^*) \leq O(\sqrt{T}), \quad (15)$$

where ℓ_t is a strongly convex function and the sequence \mathbf{p}_t and the fixed optimal portfolio \mathbf{p}^* all lie in the constraint set $\mathcal{P} = \left\{ \mathbf{p} \mid \mathbf{D}_\ell \mathbf{p} \geq 0, \mathbf{D}_s \mathbf{p} \leq 0, \mathbf{a}^\top \mathbf{p} \leq \frac{1+r}{B_\ell+r} \right\}$.

6. EXPERIMENTS AND RESULTS

6.1 Datasets

The experiments were conducted on 5 datasets with data taken from: Dow Jones Industrial Average (DJIA), New York Stock Exchange (NYSE), Standard & Poor’s 500 (S&P 500), and the Toronto Stock Exchange (TSX). The DJIA dataset [2] consists of 30 stocks and 507 trading days over a period of 2 years from 2001 to 2003. The NYSE dataset consists of 36 stocks and 5651 trading days over a period of 22 years from 1962 to 1984 and has been widely used in the literature [1, 2, 4, 13]. The S&P500^a dataset [2] consists of 25 stocks and 1276 trading days over a period of 5 years from 1998 to 2003 which includes the dot-com crash. The S&P500^b dataset¹ contains the most stocks with 263 and 505 trading days over a period of 2 years from 2007 to 2009 which includes the financial and housing crash. The TSX dataset [2] consists of 88 stocks and 1259 trading days over a period of 5 years from 1994 to 1998.

The datasets are very different in nature where 25 out of the 30 stocks (83%) in the DJIA lost value, every stock in the NYSE increased in value, 7 of the 25 stocks (28%) in the SP500^a lost value, 253 of the 263 stocks (96%) in SP500^b lost value, and 32 out of the 88 stocks (36%) in TSX lost value.

6.2 Methodology and Parameter Setting

In all our experiments we start with \$1 as our initial investment and an initial portfolio with maximum leverage uniformly distributed over all the positions. We use Algorithm 2 to obtain our portfolios sequentially and compute the transaction cost-adjusted wealth each day.

Since the five datasets are very different in nature, we experimented with various parameter values using a grid search in the following ranges: $\delta \in \{5, 10, \dots, 50\}$, and each of η, α_1, α_2 , and λ following a log-scale in $[10^{-6}, 10^3]$ to observe their affect on our portfolio and found stable behavior in these ranges. Moreover, we chose a reasonable range of transaction costs $\gamma \in [0\%, 2\%]$ to observed their affect on the transaction-cost adjusted wealth.

Typical yearly margin interest rates are between 5% and 8%. For all experiments we set the daily interest rate $r = 0.000245$ which is equivalent to a yearly interest rate of 6.3%. Rates around 6.3% have been used before in the literature [4, 12, 13]. For each dataset, we computed the parameters B_ℓ and B_s to the nearest hundredth decimal place in hindsight (Table 1). In practice, one must use historical data

¹http://www-users.cs.umn.edu/~njohnson/port_sel.html

to compute such values, thus, no-ruin guarantees can only be made probabilistically. Alternatively, one could develop online generalizations of Bayesian Optimization or suitable variants of parameter free online learning to avoid parameter setting in hindsight.

6.2.1 Constructing Structured Graph \mathcal{G}

In all our experiments, we construct the graph \mathcal{G} by fully connecting each stock so every node has degree $n - 1$. We compute their similarity by calculating the linear correlation coefficient w_{ij} between each pair of stocks i and j using the previous δ days of price relatives to get the weighted adjacency matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$. The positional graph \mathcal{G}_p was then constructed as in (2).

6.3 Cumulative Wealth

To evaluate the practical application of SHERAL, we analyze the performance and compare several benchmark algorithms with parameters tuned for maximum cumulative wealth (without transaction costs).

6.3.1 Leveraged Long and Short Effect on EG

First, we take the well-known algorithm EG [13] and observe its performance on the 5 datasets. To observe the affect that long and short positions with leverage have, we further experiment with variants of EG to allow: (1) long-only positions with leverage (LO), (2) short-only positions with leverage (SO), and (3) long and short positions with leverage (LS).

EG Variants Formulation EG allows long-only positions $\mathbf{p} \geq 0$ and uses the relative entropy function as the proximal term, i.e., $d(\mathbf{p}, \mathbf{p}_t) = \sum_{i=1}^n p(i) \log \left(\frac{p(i)}{p_t(i)} \right)$ where $\mathbf{p} \in \Delta_n = \{p(i) \geq 0 \forall i, \sum_i p(i) = 1\}$. Since we need to allow $p(i) \leq 0$ for short positions, we instead use $d(\mathbf{p}, \mathbf{p}_t) = \frac{1}{2} \|\mathbf{p} - \mathbf{p}_t\|_2^2$. Additionally, we utilize the loss function defined in (11). Essentially, the variant EG problem is

$$\min_{\mathbf{p} \in \mathcal{P}} \eta \langle \mathbf{p}, \nabla \ell_t(\mathbf{p}_t) \rangle + \frac{1}{2} \|\mathbf{p} - \mathbf{p}_t\|_2^2 \quad (16)$$

where $\mathcal{P} = \{\mathbf{p} \mid \mathbf{q}_\ell \geq 0, \|\mathbf{q}_\ell\|_1 \leq \frac{1+r}{B_\ell+r}\}$ for LO, $\mathcal{P} = \{\mathbf{p} \mid \mathbf{q}_s \leq 0, \|\mathbf{q}_s\|_1 \leq \frac{1+r}{B_s+r}\}$ for SO, and $\mathcal{P} = \{\mathbf{p} \mid \mathbf{q}_\ell \geq 0, \mathbf{q}_s \leq 0, \mathbf{a}^\top \mathbf{p} \leq \frac{1+r}{B_\ell+r}\}$ for LS. We can see that this is a special case of (12) where $\lambda = 0$. Therefore, SHERAL is able to perform at least as good as the EG* variants.

Results The results are presented in Table 2. From this table, we can see that the original EG is outperformed by at least one of the EG* variants in each dataset. Interestingly, when only adding leverage to EG, i.e. EG* (LO), the performance improves substantially for the NYSE, SP500^a, and TSX datasets. It seems that EG* (LO) is able to select the correct stocks and that increasing the investment power dramatically enhances the multiplicative gain in wealth.

	DJIA	NYSE	SP500 ^a	SP500 ^b	TSX
B_ℓ	0.60	0.26	0.31	0.61	0.64
B_s	0.21	0.36	0.25	0.67	0.94

Table 1: Table of B_ℓ and B_s values for each dataset.

	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 ⁻⁴	20.90	2.21	1.0×10³
EG* (SO)	0.63	0.04	0.34	1.10	1.07
EG* (LS)	2.00	6.6×10 ⁻⁴	20.65	2.26	1.62
SHERAL ($\lambda > 0$)	2.47	1.8×10¹⁵	19.89	7.84	8.74

Table 2: Cumulative wealth for EG, leveraged long-only (LO), short-only (SO), and long/short (LS) variants of EG*, and SHERAL with $\lambda > 0$.

Additionally, we see that EG* (SO) is able to earn more wealth for the SP500^b dataset than EG. This is because almost all stocks in this dataset lose value and EG is not able to hold short positions thereby limiting it to invest in stocks that are performing poorly. However, we also see that EG* (LO) earns more wealth than EG* (SO) even though it is limited by only holding long positions. It seems that for the few stocks that have increases in value, the leverage is enough to allow it to earn more wealth. Finally, we see that the EG* (LS) is able to outperform EG on each datasets even though they are significantly different in nature. It is also able to compete with EG* (LO) in 4 of the 5 datasets. From these results, we can see that adding leverage to EG significantly increases cumulative wealth. Moreover, it seems that short positions do not have a strong impact on cumulative wealth but they do allow for more flexibility and reasonable performance on datasets with both bull and bear markets.

We can observe the affect that structured hedging has on wealth by comparing the cumulative wealth of SHERAL to that of EG and EG* variants. We can especially see its affect by looking at EG* (LS) since it is equivalent to setting $\lambda = 0$ in (12). We can see that SHERAL is able to earn more wealth than EG and all EG* variants on 3 of the 5 datasets. This implies that there is some volatility in these datasets and the structured hedging helps alleviate this volatility. The structured hedging has a positive affect on most datasets and, as we will see later, it is able to reduce the risk and earn similar amounts of wealth. When $\lambda = 0$, SHERAL can earn as much as EG and all EG* variants so for datasets where hedging is not beneficial, we can still earn the same amount of wealth.

In Figure 1, we can see how the transaction-cost adjusted cumulative wealth changes as we vary the structured hedging weight λ in comparison with the best performing variant EG* (LO) for the NYSE dataset. SHERAL is able to earn more wealth even with reasonable transaction costs than the best EG variant (which does not include transaction costs).

6.3.2 Comparison with Benchmark Algorithms

In addition to a comparison between EG and our EG* variants, we compared the performance between several benchmark algorithms: a buy-and-hold strategy (U-BAH), a uniform constant rebalanced portfolio (U-CRP), Universal Portfolios (UP) [4], Online Lazy Updates (OLU) [7], and the best single stock in hindsight. These algorithms were designed to only invest in long positions and without leverage.

Further, we again compared against several variants of these algorithms: a long-only, short-only, and long/short with leverage variant of both U-BAH and U-CRP, and a long-only and short-only with leverage variant of the best single stock. The results of these algorithms and variants on the 5 datasets are presented in Table 3.

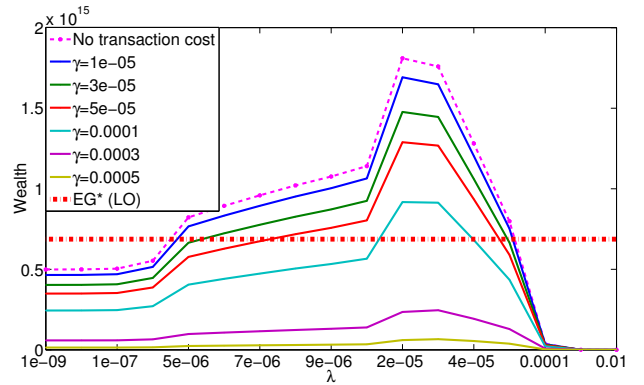


Figure 1: Transaction cost-adjusted wealth for the NYSE dataset with varying values of λ . SHERAL returns more wealth than the best competing algorithm even with transaction costs.

We can see that out of those algorithms that only invest in long positions and without leverage, the best single stock tends to outperform the best with OLU being competitive in most cases. We can see how the market performs by looking at the U-BAH algorithm. If U-BAH returns < 1 then the market was down for that dataset. For such datasets (DJIA, SP500^b), we see that both U-CRP and UP perform about the same and lose money. However, the best stock earns money for these datasets. We can see that there is at least one stock that performs well even if the majority of the stocks do not. OLU is able to identify this stock, or the few similar stocks, for SP500^b and earn money but is not able to do the same for DJIA. Comparing the U-BAH and U-CRP variants on the SP500^b dataset, we see that only U-BAH* (SO) and (LS) are able to earn money. Since for this dataset, the vast majority of the stocks decrease in value, these variants are able to hold short positions and take advantage of this. U-CRP* (SO) and (LS) lose money but still perform better than the (LO) variant for this dataset.

For the NYSE dataset, we can see that U-CRP* (LO) earns significantly more wealth than all the other algorithms (except SHERAL). Since all the stocks in the NYSE dataset increase in value, holding long positions is beneficial and

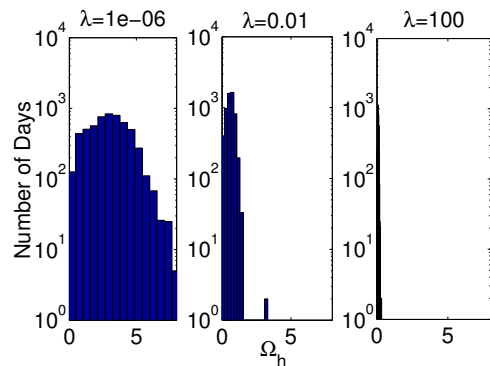


Figure 2: Total value of the hedging penalty Ω_h with varying λ for the NYSE dataset. As λ increases, the value of the hedging penalty function Ω_h decreases.

	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	3.68×10^{-6}	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	1.04×10^{-6}	1.00×10^{-2}	0.97	0.54
U-CRP* (LS)	0.83	0.05	0.45	0.93	0.92
Best Stock* (LO)	1.09 (P&G)	65.71 (PM)	0.56 (WMT)	1.15 (SWN)	7.84 (GTA)
Best Stock* (SO)	1.13 (MCD)	1.45×10^{-5} (DD)	0.02 (KO)	4.71 (GCI)	2.11 (IFP)
SHERAL ($\lambda > 0$)	2.47	1.81×10^{15}	19.89	7.84	8.74

Table 3: Cumulative wealth (without transaction costs) of SHERAL, benchmark algorithms, and several variants for each of the five datasets.

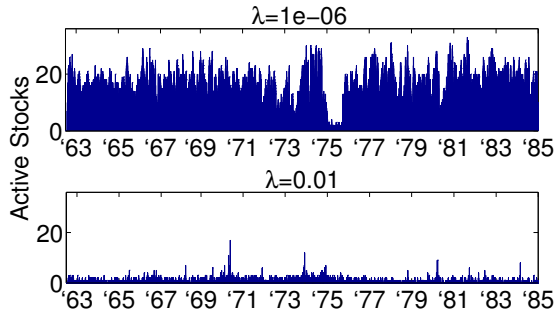


Figure 3: Active stocks with varying λ . As λ increases the number of active stocks decreases which indicates either more hedging or less total investing.

adding leveraging provides even more earning power. However, comparing the leveraged (LO) best stock with the non-leveraged (LO) best stock, we can see that for each dataset, the leverage does not appear to help that much and even in some cases hurts the cumulative wealth. This is because leverage magnifies not only gains but also losses. We can see the benefits of leverage but also the drawbacks. If we have a portfolio that is allowed to switch between stocks with leverage, there can be huge gains. However, if we are stuck investing in a single stock with leverage, there are few gains and even some losses.

Looking at the last row of Table 3, we observe the results for our SHERAL algorithm. We can see that SHERAL is able to outperform all other algorithms for each of the datasets even though they are very different in nature. For example, SHERAL earns \$7.84 on the SP500^b dataset where 96% of the stocks decrease in value and is also able to earn $\$1.81 \times 10^{15}$ on the NYSE dataset where all stocks increase in value. This shows the flexibility and power of being able to invest in both long and short positions with leverage.

6.4 Effect of the hedging function Ω_h and λ

The hedging penalty function Ω_h encourages structural hedging between assets. The amount of hedging is further controlled by the value of λ which has an affect on (a) the

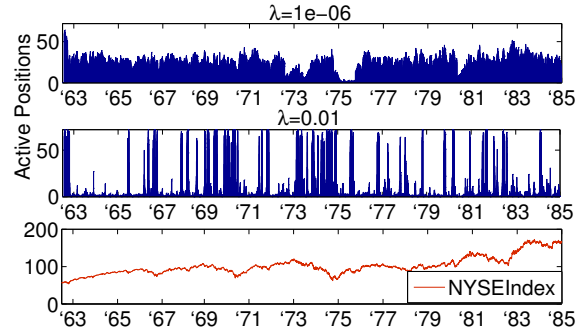


Figure 4: Active positions with varying λ . When the NYSE Index decreases, hedging increases with $\lambda = 0.01$.

value of Ω_h each day, (b) the number of active stocks, and (c) the number of active positions.

(a) **Value of Ω_h** With the hedging penalty function Ω_h , we are encouraging different levels of asset hedging depending on the value of λ . From Figure 2, we can see the affect λ has on $\Omega_h = \mathbf{p}^\top \mathbf{L} \mathbf{p}$ with $\eta = 0.1$, $\alpha_1 = 0.1$, and $\alpha_2 = 0.1$ for the NYSE dataset with $\delta = 20$ days. With a low λ value of 10^{-6} , there are many days with large values of Ω_h . As we increase λ to 100, we see that the value of Ω_h becomes very small with most days having a value of 0.

(b) **Number of Active Stocks** An active stock is a stock which has a significant percent of the wealth, e.g, 1%, 10%, etc., invested in it between both long and short positions. For instance, if there is 1% of the wealth invested in the long position and 1% invested in the short position, the total *effective wealth* invested is 0% since the amounts are equal and cancel out. An active stock has a significant amount of total effective wealth invested. It is a measure of the number of non-hedged assets where a higher number indicates less hedging and a lower number indicates more hedging or less total investing, i.e., more wealth held in the bank.

From Figure 3, we can see that with $\eta = 0.1$, $\alpha_1 = 1$, $\alpha_2 = 10$, and $\lambda = 10^{-6}$, the number of active stocks for the

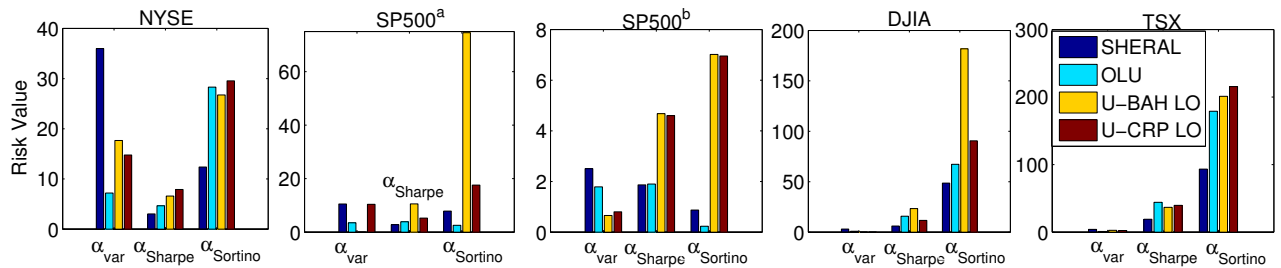


Figure 5: Average risk for each algorithm and dataset with optimal parameters in terms of wealth returned. SHERAL computes portfolios with less risk than U-BAH (LO) and U-CRP (LO) for almost all risk measures and datasets and is competitive with the non-leveraged algorithm OLU.

NYSE dataset is high, with around 25 out of 36 stocks active (about 70%) on average. With the other parameters fixed and only increasing λ to 0.01, the number of active stocks decreases to around 4 (about 11%). This could be due to two reasons: (1) the amount invested in both long and short positions increases and the total effective wealth invested decreases, or (2) the total amount invested decreases with the rest kept in the bank.

(c) Number of Active Positions We define an active position in a similar way to that of an active stock, however we consider the positions independent of one another. If both the long and short positions have the same and significant amount invested in them, then both are considered active. From Figure 4, we can see the affect λ has on the number of active positions for the same parameters values as that of Figure 3. With a low λ value of 10^{-6} , we can see that the number of active positions is reasonably high, with around the same amount as the number of active stocks in Figure 3 (note the scale differences). This shows that when λ is small for these set of parameters, SHERAL is not hedging much. When λ increases to 0.01, the number of active positions increases some days and decreases other days.

Comparing this to the NYSE index value in the bottom plot, we can see that the days where SHERAL is hedging more corresponds to days that the index decreases, e.g. 1969-1970, 1973-1975, and 1981-1983. This is explained by the fact that with this set of parameters, we are emphasizing long positions more even though $\alpha_2 > \alpha_1$ because of the difference in scale. For example, the mean price relative for the NYSE dataset is 1.0006, and for a long position the gain is $\mathbf{q}_l^\top \mathbf{x}_t$ which on average is of the order $1 \times \mathbf{q}_l$ whereas the gain for short positions is $\mathbf{q}_s^\top (\mathbf{x}_t - 1 + r)$ which on average is of the order $(10^{-4} + r) \times \mathbf{q}_s$. With $r = 0.000245$, to weight the positions equally we would have to set α_2 to be in the range $[10^3, 10^4]$. However, for these results $\alpha_2 = 10$ so the long positions are emphasized more. Since there is more hedging on days the market decreases, this indicates that SHERAL is trying to take advantage of increasing stocks on some days whereas other days, when the market is crashing, it is not quite sure which stocks will perform well so it is hedging more between them.

6.5 Risk Comparison

From Section 6.3, we saw that SHERAL is able to return more wealth than all of the state-of-the-art algorithms on all of the datasets, and more wealth than EG and our EG* variants in three out of the five datasets. However, as

Markowitz postulated, we should seek low risk in addition to high returns. As such, we compare the risk exposure for each of the competing algorithms with optimal parameters with respect to wealth using three common measures of risk: (a) covariance, (b) Sharpe ratio, and (c) Sortino ratio.

(a) Covariance We compute the covariance Σ_t using the previous δ days of price relatives. We measure the risk of a portfolio \mathbf{p}_t w.r.t. a uniform constant rebalanced portfolio \mathbf{u} as $\alpha_{var} = \mathbf{p}_t^\top \Sigma_t \mathbf{p}_t / \mathbf{u}^\top \Sigma_t \mathbf{u}$. High α_{var} implies high risk and low α_{var} implies low risk.

(b) Sharpe ratio The Sharpe ratio [16] measures how much the return (percent gain or loss on investment) of a portfolio compensates for the level of risk taken. It computes what can be considered as a risk-adjusted return for a given portfolio and benchmark return. It does this by measuring both the downwards and upwards volatility. A higher Sharpe ratio implies better compensation for the risk exposure. We compute the Sharpe ratio of a portfolio as $\alpha_{Sharpe} = (R - R_b) / \sqrt{\text{var}(R - R_b)}$ where R is the return for the portfolio and R_b is the benchmark return which is typically a large index such as the S&P500.

(c) Sortino ratio The Sortino ratio [17] is similar to the Sharpe ratio, however it only measures the downwards volatility. Typically, upwards volatility is encouraged as we would gladly accept the price of a stock we have invested long in to go up. However, the Sharpe ratio penalizes this type of volatility where the Sortino ratio does not. We compute the Sortino ratio as $\alpha_{Sortino} = (R - R_b) / DR$ where DR is the standard deviation of negative returns (losses).

To be consistent in the plots and have each bar represent the level of risk exposure, we have plotted the negative Sharpe and Sortino ratios since a low ratio implies a high risk relative to the return. Therefore, for each of the bar plots in Figure 5, a higher bar height implies higher risk. Additionally, we compare algorithms that are not special cases of SHERAL, i.e., EG or EG variants. We also only compare the best performing algorithms and variants with parameters tuned for optimal wealth for each algorithm.

From Figure 5, we can see that SHERAL is competitive with OLU in terms of risk, and in many cases having less risk. SHERAL consistently has less risk than U-BAH (LO) and U-CRP (LO). These results are encouraging because in addition to earning more wealth, SHERAL is able to reduce risk in spite of using leverage which inherently increases risk.

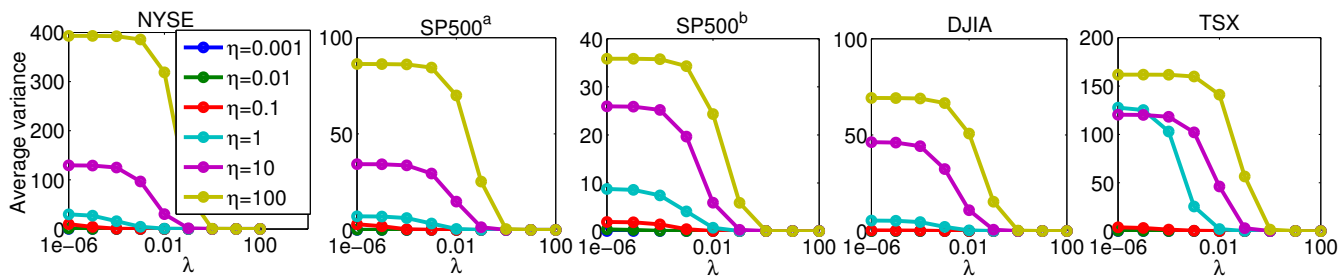


Figure 6: Average α_{var} risk for each dataset with varying η and λ . With a higher η value, the average variance risk is higher, however as we increase λ the risk decreases for each η value across all datasets.

Additionally, for algorithms which do use leverage, U-BAH (LO) and U-CRP (LO), SHERAL is exposed to much less risk than these algorithms and earns more wealth.

6.6 Risk and λ

Even though our hedged resource allocation with leverage framework (8) does not explicitly take risk into account, we can control it by setting the value of λ . We observe how the value of λ affects the amount of risk our portfolios are exposed to using variance as the measure of risk. We do not consider the Sharpe or Sortino ratios here since with varying parameters, the return in wealth changes drastically and the risk-adjusted returns from the Sharpe and Sortino ratios cannot effectively illustrate how only the risk changes.

From Figure 6, we can see that across each dataset the behavior is consistent. As we increase η , the average variance is higher across all values of λ . As we increase λ , the average variance decreases to zero for all η values. This illustrates how as we increase λ , we are encouraging more hedged portfolios and reducing risk. Thus, we can effectively control the level of risk by setting the value of λ .

7. CONCLUSIONS

In this paper, we have developed a framework and algorithm for hedging online resource allocations with leverage (SHERAL) and showed how it can be applied to the problem of online portfolio selection. Our analysis shows that SHERAL is competitive with reasonable fixed strategies which have the power of hindsight. Our experimental results illustrate the affect long and short positions and a graph-based hedging penalty function Ω_h have on the cumulative wealth and various measures of risk. We exhaustively show how SHERAL compares in terms of wealth with state-of-the-art portfolio selection algorithms on five datasets. Additionally, we experimented with variants of such algorithms to allow long and short positions with leverage and showed how SHERAL is competitive with these variants in terms of risk and wealth even with reasonable transaction costs.

Acknowledgements

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.

8. REFERENCES

[1] A. Agarwal, E. Hazan, S. Kale, and R. Schapire. Algorithms for portfolio management based on the newton method. In *ICML*, pages 9–16, 2006.

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

[3] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[4] T. Cover. Universal portfolios. *Mathematical Finance*, 1:1–29, 1991.

[5] P. Das and A. Banerjee. Meta optimization and its application to portfolio selection. In *KDD*, 2011.

[6] P. Das and A. Banerjee. Online quadratically constrained convex optimization with applications to risk adjusted portfolio selection. Technical report, University of Minnesota, 2012.

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

[8] P. Das, N. Johnson, and A. Banerjee. Online portfolio selection with group sparsity. In *AAAI*, 2014.

[9] N. R. Devanur and T. P. Hayes. The adwords problem: Online keyword matching with budgeted bidders under random permutations. In *EC*, pages 71–78, 2009.

[10] S. Ermon, J. Conrad, C. Gomes, and B. Selman. Risk-sensitive policies for sustainable renewable resource allocation. In *IJCAI*, pages 1942–1948, 2011.

[11] D. Golovin, A. Krause, B. Gardner, S. J. Converse, and S. Morey. Dynamic resource allocation in conservation planning. In *AAAI*, 2011.

[12] L. Györfi, G. Ottucsak, and H. Walk. *Machine Learning for Financial Engineering*. Imperial College Press, London, UK, UK, 2011.

[13] D. Helmbold, E. Schapire, Y. Singer, and M. Warmuth. On-line portfolio selection using multiplicative weights. *Mathematical Finance*, 1998.

[14] N. Johnson and A. Banerjee. Online resource allocation with structured diversification. In *SDM*, 2015.

[15] R. Munos. Efficient resources allocation for markov decision processes. In *NIPS*, pages 1571–1578, 2001.

[16] W. Sharpe. Mutual fund performance. *Journal of Business*, pages 119–138, 1966.

[17] F. Sortino and R. van der Meer. Downside risk. *Journal of Portfolio Management*, pages 27–31, 1991.

[18] S. Xue, A. Fern, and D. Sheldon. Dynamic resource allocation for optimizing population diffusion. In *AISTATS*, pages 1033–1041, 2014.