Online Resource Allocation with Structured Diversification

Nicholas Johnson* Arindam Banerjee*

Abstract

A variety of modern data analysis problems, ranging from finance to job scheduling, can be considered as online resource allocation (ORA) problems. A key consideration in such ORA problems is some notion of risk, and suitable ways to alleviate risk. In several settings, the risk is structured so that groups of assets, such as stocks, are exposed to similar risks. In this paper, we present a formulation for online resource allocation with structured diversification (ORASD) as a constrained online convex optimization problem. The key novel component of our formulation is a constraint on the $L_{(\infty,1)}$ group norm of the resource allocation vector, which ensures that no single group gets a large share of the resource and, unlike $L_{(1,p)}$ norms used for overlapping group Lasso, does not impose sparsity structures over groups. We instantiate the problem in the context of portfolio selection, propose an efficient ADMM algorithm, and illustrate the effectiveness of the formulation through extensive experiments on two benchmark datasets.

1 Introduction

A variety of modern data analysis problems can be considered as resource allocation problems. The problems range from investment of resources to a variety of assets, such as investing money in the stock market or assigning teams to software projects, to modern systems level challenges, such as job scheduling in compute clusters or data/content storage and delivery over the internet. 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, changes in demand for data/content, etc.

A key consideration in such online resource allocation (ORA) problems is some notion of risk, and suitable ways to diversify the allocation to alleviate risk. For portfolio selection, putting all of ones money in one or a few stocks is considered risky, since those few stocks may not perform well over time. Further, companies usually invest in multiple projects, rather than one or a few, since some of those projects will succeed and several others may fail. One considers similar notions of risk and diversification strategies for systems level challenges, such as fault-tolerant designs for compute job scheduling and provisioning for data/content delivery.

In several such ORA settings, the risk is often structured and calls for structured diversification strategies. For example, in portfolio selection, one considers market sectors, such as energy, technology, utilities, etc., which gives a structure beyond individual stocks. Often, stocks within a sector move together in response to external influences so that investing in just one sector can be risky. A structurally diverse portfolio would invest in multiple sectors to alleviate risk.

In this paper, we present a novel formulation for online resource allocation with structured diversification (ORASD). The formulation considers groups over the assets of interest, such as stocks, where the groups can be of different sizes and they can overlap. We also outline a way in which these groups can be inferred from the data, e.g., based on their correlation structures. The key novel component of our formulation is the $L_{(\infty,1)}$ group norm, which is rather different from $L_{(1,p)}$ type group norms typically used for overlapping group Lasso and related problems. We pose the ORASD problem as a suitable online convex optimization problem with the constraint that the $L_{(\infty,1)}$ group norm of the resource allocation vector has to be bounded within a prespecified limit. Such a constraint ensures that no single group gets a large share of the resource. Further, unlike overlapping group Lasso, there is no sparsity restriction on the resource allocation vector, so that one can invest in all assets if that helps the resource allocation objective. We instantiate the problem in the context of portfolio selection and propose an efficient algorithm based on the alternating direction method of multipliers (ADMM). We illustrate the effectiveness through extensive experiments on two benchmark datasets and several baselines from the existing literature.

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

*Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455 {njohnson,banerjee}@cs.umn.edu

2 Related Work

There is a large literature on resource allocation with many papers posing the problem as an auction [11], mechanism design problem [2], Markov Decision Process [12, 24], matching problem [15], and planning problem [16, 27]. However, few have considered risk when computing allocations in an online fashion [14, 18, 19]. Additionally, little work in online portfolio selection have considered risk in their algorithmic setting [8]. However, previous work [1, 5, 6, 7, 9, 10, 17] have shown that 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.

3 Diversified Resource Allocation

In this section, we introduce a framework for resource allocation with structured diversity, and consider the online resource allocation problem under such structured diversity. In Section 4, we consider the online portfolio selection problem as an instance of such diversified resource allocation.

3.1 Structured Diversity with $L_{(\infty,1)}$ Norm We consider a resource allocation problem over n objects, where the goal is find a probability distribution $\mathbf{p} \in \Delta_n$, the n-dimensional probability simplex, which determines how to split up a resource over the n objects such that a certain (convex) objective $f(\mathbf{p})$ is minimized. For example, in the context of investment (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 resources (say, money) should one put on each asset (stock). The basic idea of diversification is to avoid putting all of the resources on one asset. A simple way to accomplish this is to put a cap or upper bound on the amount of resources which can be put on any asset, i.e., $p(i) \leq \kappa$ for some $\kappa \in [0, 1]$. A potential issue with such an approach is that there may be structural dependencies between the assets. For example, in the context of portfolio selection, selling some Google stocks to buy Apple stocks may not accomplish the goals of diversification when the entire tech sector goes down. In this example, Google and Apple as assets can be considered structurally related, both being part of the tech sector. The goal of structured diversification is to develop diversification strategies which explicitly consider such structurally related groups and diversify across such groups.

For the development, we assume knowledge of such structurally related groups, and outline approaches for inferring such groups directly from the data in Section 4 in the context of portfolio selection. Let $\mathcal{G} = \{g_1, \ldots, g_m\}$ be the set of groups, where g_i is the set of indices for the n_i assets in the group. The groups can be of different sizes, $|g_i| = n_i$, and the groups may overlap, i.e., $g_i \cap g_j \neq \emptyset$.

Given such group structure, we introduce the $L_{(\infty,1)}$ group norm which plays a key role in structured diversity. For any $\mathbf{x} \in \mathbb{R}^n$ and a set of (possibly overlapping, different sized) groups $\mathcal{G}$, the $L_{(\infty,1)}$ group norm is defined as:

$$\|\mathbf{x}\|^{\mathcal{G}}_{(\infty,1)} = \left\| \left[\|\mathbf{x}_{g_1}\|_1 \ldots \|\mathbf{x}_{g_m}\|_1 \right]^\top \right\|_\infty \tag{3.1}$$

where $\mathbf{x}_{g_i}$ is a vector of length n with value equal to $\mathbf{x}$ for indices in g_i and 0 otherwise. The $L_{(\infty,1)}$ group norm is determined by the largest L_1 norm over all groups in $\mathcal{G}$. This norm is rather different in character compared to the $L_{(1,2)}$ norm [21], used in the context of overlapping group Lasso and multi-task learning, as well as the $L_{(1,\infty)}$ group norm [23]. The $L_{(1,p)}$ group norms accomplish sparsity over the groups, i.e., certain groups can become all zeroes. Such a structure is not desirable in the context of diversified portfolio selection, since one wants to distribute the resource to multiple groups for diversification. In fact, for $\|\mathbf{x}\|^{\mathcal{G}}_{(\infty,1)} \leq \kappa$, each individual group g_i has the flexibility of increasing their L_1 norm to κ without changing the norm ball (penalty). Similarly, one could also consider any $L_{(\infty,p)}$ group norm, which looks at the maximum over L_p norms over groups. In the context of resource allocation, since the key object of interest is a probability distribution, we work with the $L_{(\infty,1)}$ group norm in this paper. Thus, for the purposes of structured diversity in resource allocation, we will consider problems of the form

$$\min_{\mathbf{p} \in \Delta_n} \; f(\mathbf{p}) \quad \text{s.t.} \quad \|\mathbf{p}\|^{\mathcal{G}}_{(\infty,1)} \leq \kappa \; , \tag{3.2}$$

for a suitably chosen $\kappa \in [0, 1]$.

3.2 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, $\mathbf{p}_t \in \Delta_n$, without knowing $f_t(\cdot)$ and incur a loss of $f_t(\mathbf{p}_t)$. In addition to being feasible, we add two additional requirements on $\mathbf{p}_t$: (1) $\mathbf{p}_t$ needs to stay close to $\mathbf{p}_{t-1}$ since we do not want the resource allocation to change drastically in every time step, which may have a cost associated with it, and (2) $\|\mathbf{p}_t\|^{\mathcal{G}}_{(\infty,1)} \leq \kappa$ so that structural diversity is maintained over time. Thus, the sub-problem at time t

takes the form

$$\text{(3.3)} \quad \min_{\mathbf{p}\in\Delta_n} \ \eta\,\ell_t(\mathbf{p}) + \Omega(\mathbf{p},\mathbf{p}_{t-1}) \quad \text{s.t. } \|\mathbf{p}\|^{\mathcal{G}}_{(\infty,1)} \leq \kappa \ ,$$

where ℓ_t denotes a suitable resource allocation loss and $\Omega(\cdot,\cdot)$ is a convex penalty function based on the change in $\mathbf{p}_t$. Ideally, over T rounds we would like to minimize the constrained cumulative loss

$$\text{(3.4)} \quad \sum_{t=1}^{T} \eta\,\ell_t(\mathbf{p}_t) + \Omega(\mathbf{p}_t,\mathbf{p}_{t-1}) \quad \text{s.t. } \|\mathbf{p}_t\|^{\mathcal{G}}_{(\infty,1)} \leq \kappa \ .$$

However, in the online setting, absolute minimization of (3.4) 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, i.e.,

$$\text{(3.5)} \qquad R_T = \sum_{t=1}^{T} f_t(\mathbf{p}_t) - \min_{\mathbf{p}^*} \sum_{t=1}^{T} f_t(\mathbf{p}^*) \leq o(T)$$

where $f_t(\mathbf{p}){=}\eta\ell_t(\mathbf{p}){+}\Omega(\mathbf{p},\mathbf{p}_{t-1})$. 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 the problem obtained by a first-order Taylor expansion of f_t at $\mathbf{p}_t$ along with a proximal term, so that

$$\text{(3.6)} \qquad \mathbf{p}_{t+1} = \underset{\substack{\mathbf{p}\in\Delta_n \\ \|\mathbf{p}\|^{\mathcal{G}}_{(\infty,1)}\leq\kappa}}{\operatorname{argmin}} \ \eta\langle \mathbf{p}, \nabla\ell_t(\mathbf{p}_t)\rangle + d(\mathbf{p},\mathbf{p}_t) \ ,$$

where $d(\mathbf{p},\mathbf{p}_t) = \frac{1}{2}\|\mathbf{p}-\mathbf{p}_t\|_2^2 + \lambda\Omega(\mathbf{p},\mathbf{p}_t)$ and the parameters $\eta,\lambda \geq 0$. As we discuss in Section 4, one can extend existing arguments to obtain a sub-linear regret bound for such online resource allocation.

4 Diversified Online Portfolio Selection

Given the resource allocation with structured diversity framework, we now introduce online portfolio selection and propose a primal-dual algorithm for solving (3.6) in this context.

4.1 Online Portfolio Selection We consider a stock market consisting of n stocks $\{s_1,\ldots,s_n\}$ 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 $\mathbf{x}_t = [x_t(1),\ldots,x_t(n)]^\top$ denote the vector of price relatives for day t, and let $\mathbf{x}_{1:t}$ denote the collection of such price relative vectors up to and including day t. A portfolio $\mathbf{p}_t = [p_t(1),\ldots,p_t(n)]^\top \in \Delta_n$ on day t can be viewed as a probability distribution over n stocks that prescribes investing $p_t(i)$ fraction of the current wealth 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. The multiplicative gain in wealth at the end of day t is then $\mathbf{p}_t^\top\mathbf{x}_t$. For a sequence of price relatives $\mathbf{x}_{1:t-1} = \{\mathbf{x}_1,\ldots,\mathbf{x}_{t-1}\}$ 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 $\mathbf{p}_t^\top\mathbf{x}_t$. Over T periods, for a sequence of portfolios $\mathbf{p}_{1:T} = \{\mathbf{p}_1,\ldots,\mathbf{p}_T\}$, the multiplicative gain in wealth is $S_T(\mathbf{p}_{1:T},\mathbf{x}_{1:T}) = \prod_{t=1}^{T}\left(\mathbf{p}_t^\top\mathbf{x}_t\right)$ and the logarithmic gain in wealth is $LS_T(\mathbf{p}_{1:T},\mathbf{x}_{1:T}) = \sum_{t=1}^{T}\log\left(\mathbf{p}_t^\top\mathbf{x}_t\right)$.

Ideally, for a costless environment (no transaction costs) we would like to maximize $LS_T(\mathbf{p}_{1:T},\mathbf{x}_{1:T})$ over $\mathbf{p}_{1:T}$. However, online portfolio selection cannot be posed as a batch optimization problem due to the temporal nature of the choices: $\mathbf{x}_t$ is not available when one has to decide on $\mathbf{p}_t$. Further, (statistical) assumptions regarding $\mathbf{x}_t$ can be difficult to make.

4.2 ORASD Algorithm Online portfolio selection can now be viewed as a special case of our online resource allocation with structured diversity setting where $\ell_t(\mathbf{p}_t) = -\log(\mathbf{p}_t^\top\mathbf{x}_t)$, and $\Omega = \|\mathbf{p}-\mathbf{p}_t\|_1$. The L_1 penalty term on the difference of two consecutive portfolios measures the fraction of wealth traded and encourages lazy updates to the portfolio to limit transaction costs. The parameter λ controls the amount that can be traded every day. The $L_{(\infty,1)}$ penalty term forces the portfolio to spread out the investment amongst the groups. The level of diversification depends on the value of κ. Additionally, since we allow overlapping groups and will solve (3.6) via lifting, we need to add a consensus constraint $\mathbf{p}_{g_i} = \mathbf{S}_{g_i}\mathbf{p}\ \forall i$ where $\mathbf{p}_{g_i}$ is a vector of length n with value equal to $\mathbf{p}$ for indices in g_i and 0 otherwise and $\mathbf{S}_{g_i}$ is a known diagonal matrix with $\mathbf{S}_{g_i}(j,j) = 1$ if element j is in group g_i and 0 otherwise. The online portfolio selection with structured diversification problem is now

$$\text{(4.7)} \quad \min_{\substack{\mathbf{p}\in\Delta_n \\ \|\mathbf{p}\|^{\mathcal{G}}_{(\infty,1)}\leq\kappa \\ \mathbf{S}_{g_i}\mathbf{p}=\mathbf{p}_{g_i}\ \forall i}} \ \eta\langle \mathbf{p}, \nabla\ell_t(\mathbf{p}_t)\rangle + \frac{1}{2}\|\mathbf{p}-\mathbf{p}_t\|_2^2 + \lambda\|\mathbf{p}-\mathbf{p}_t\|_1 \ .$$

(4.7) consists of smooth and non-smooth terms in the

Algorithm 1 ORASD Algorithm with ADMM

Input $\mathbf{p}_t, \mathbf{x}_t, \mathbf{S}_{g_{1,\dots,g_m}}, \eta, \lambda, \beta$
Initialize $\mathbf{p}, \mathbf{p}_{g_i}, \mathbf{z}, \mathbf{u}_i, \mathbf{v} \in \mathbf{0}^n, k = 0$
Set $\hat{\mathbf{a}} = \left((1+\beta)I + \beta\left(\mathbf{S}_{g_1} + \cdots + \mathbf{S}_{g_m}\right)\right)^{-1}$
ADMM iterations
(4.9)

$$\mathbf{p}^{k+1} = \prod_{\Delta_n}\left(\hat{\mathbf{a}}\frac{\eta\mathbf{x}_t}{\mathbf{p}_t^\top\mathbf{x}_t} + \hat{\mathbf{a}}(1+\beta)\mathbf{p}_t + \hat{\mathbf{a}}\beta\Big(\mathbf{S}_{g_1}(\mathbf{p}_{g_1}^k - \mathbf{u}_1^k) + \ldots + \mathbf{S}_{g_m}(\mathbf{p}_{g_m}^k - \mathbf{u}_m^k) + \mathbf{z}^k - \mathbf{v}^k\Big)\right)$$

(4.10) $$\mathbf{p}_{g_i}^{k+1} = \prod_{\|\cdot\|_1 \le \kappa}\left(\mathbf{S}_{g_i}\mathbf{p}^{k+1} + \mathbf{u}_i^k\right) \forall i$$

(4.11) $$\mathbf{z}^{k+1} = S_{\lambda/\beta}(\mathbf{p}^{k+1} - \mathbf{p}_t + \mathbf{v}^k)$$

(4.12) $$\mathbf{u}_i^{k+1} = \mathbf{u}_i^k + \mathbf{S}_{g_i}\mathbf{p}^{k+1} - \mathbf{p}_{g_i}^{k+1} \ \forall i$$

(4.13) $$\mathbf{v}^{k+1} = \mathbf{v}^k + \mathbf{p}^{k+1} - \mathbf{p}_t - \mathbf{z}^{k+1}$$

where $\prod_{\Delta_n}$ is the projection to the simplex and S_ρ is the shrinkage operator.
Continue until **Stopping Criteria** is satisfied

objective with $L_{(\infty,1)}$ and linear constraints. We can use ADMM to solve this problem by introducing auxiliary variable $\mathbf{z} = \mathbf{p} - \mathbf{p}_t$ and moving the inequality constraint into the objective function if we let $h(\mathbf{p}_{g_i}) = \mathbb{1}_{(\|\mathbf{p}_{g_i}\|_1 \le \kappa)}$ where $\|\mathbf{p}\|_{(\infty,1)}^{\mathcal{G}} \le \kappa \equiv \|\mathbf{p}_{g_i}\|_1 \le \kappa \ \forall i$

(4.8)

$$\min_{\substack{\mathbf{p}\in\Delta_n \\ \mathbf{S}_{g_i}\mathbf{p}=\mathbf{p}_{g_i} \ \forall i \\ \mathbf{p}-\mathbf{p}_t=\mathbf{z}}} \eta\langle\mathbf{p}, \nabla\ell_t(\mathbf{p}_t)\rangle + \frac{1}{2}\|\mathbf{p}-\mathbf{p}_t\|_2^2 + \lambda\|\mathbf{z}\|_1 + \sum_{i=1}^m h(\mathbf{p}_{g_i}).$$

The ADMM formulation in (4.8) naturally lets us decouple the non-smooth terms from the smooth terms, which is computationally advantageous. The augmented Lagrangian for (4.8) is

(4.14)

$$\begin{aligned} &L(\mathbf{p}, \mathbf{p}_{g_1\ldots g_m}, \mathbf{z}, \mathbf{u}_{1\ldots m}, \mathbf{v}) = \\ &\eta\langle\mathbf{p}, \nabla\ell_t(\mathbf{p}_t)\rangle + \frac{1}{2}\|\mathbf{p}-\mathbf{p}_t\|_2^2 + \lambda\|\mathbf{z}\|_1 + \sum_{i=1}^m h(\mathbf{p}_{g_i}) \\ &+ \frac{\beta}{2}\sum_{i=1}^m \|\mathbf{S}_{g_i}\mathbf{p} - \mathbf{p}_{g_i} + \mathbf{u}_i\|_2^2 + \frac{\beta}{2}\|\mathbf{p} - \mathbf{p}_t - \mathbf{z} + \mathbf{v}\|_2^2 \end{aligned}$$

where $\mathbf{u}$ and $\mathbf{v}$ are scaled dual variables. ADMM consists of the following iterations

(4.15)

$$\begin{aligned} \mathbf{p}^{k+1} := \operatorname*{argmin}_{\mathbf{p}\in\Delta_n} \ & \eta\langle\mathbf{p}, \nabla\ell_t(\mathbf{p}_t)\rangle + \frac{1}{2}\|\mathbf{p}-\mathbf{p}_t\|_2^2 \\ & + \frac{\beta}{2}\sum_{i=1}^m \|\mathbf{S}_{g_i}\mathbf{p} - \mathbf{p}_{g_i}^k + \mathbf{u}_i^k\|_2^2 + \frac{\beta}{2}\|\mathbf{p} - \mathbf{p}_t - \mathbf{z}^k + \mathbf{v}^k\|_2^2 \end{aligned}$$

Algorithm 2 Diversified Online Portfolio Selection

Input η, λ, β; Transaction cost γ, Days lag num_{lag}
Initialize $\mathbf{p}_0(i) = \frac{1}{n}$ $i = 1, \ldots, n$, $S_0^\gamma = \$1$
For $t = 1, \ldots, T$
  Receive $\mathbf{x}_t$, the vector of price relatives
  Compute wealth: $S_t^\gamma = S_{t-1}^\gamma\left(\mathbf{p}_t^T\mathbf{x}_t - \gamma\|\mathbf{p}_t - \mathbf{p}_{t-1}\|_1\right)$
  If $t \le num_{lag}$
    $\mathbf{p}_{t+1}(i) = \frac{1}{n}$ $i = 1, \ldots, n$ (uniform portfolio)
  Else
    Compute groups: $\mathcal{G}$
    Update: $\mathbf{p}_{t+1} = \text{ORASD}(\mathbf{p}_t, \mathbf{x}_t, \mathcal{G}, \eta, \lambda, \beta)$
end for

(4.16) $$\mathbf{p}_{g_i}^{k+1} := \operatorname*{argmin}_{\mathbf{p}_{g_i}} h(\mathbf{p}_{g_i}) + \frac{\beta}{2}\|\mathbf{S}_{g_i}\mathbf{p}^{k+1} - \mathbf{p}_{g_i} + \mathbf{u}_i^k\|_2^2 \ \forall i$$

(4.17) $$\mathbf{z}^{k+1} := \operatorname*{argmin}_{\mathbf{z}} \lambda\|\mathbf{z}\|_1 + \frac{\beta}{2}\|\mathbf{p}^{k+1} - \mathbf{p}_t - \mathbf{z} + \mathbf{v}^k\|_2^2$$

(4.18) $$\mathbf{u}_i^{k+1} := \mathbf{u}_i^k + \mathbf{S}_{g_i}\mathbf{p}^{k+1} - \mathbf{p}_{g_i}^{k+1} \ \forall i$$

(4.19) $$\mathbf{v}^{k+1} := \mathbf{v}^k + \mathbf{p}^{k+1} - \mathbf{p}_t - \mathbf{z}^{k+1}\ .$$

p-update: We solve for $\mathbf{p}$ be taking the gradient of (4.15) *w.r.t.* $\mathbf{p}$ and setting it to zero to get the closed form update of $\mathbf{p}$ as

(4.20)

$$\mathbf{p} = \prod_{\Delta_n}\Big(\hat{\mathbf{a}}\frac{\eta\mathbf{x}_t}{\mathbf{p}_t^\top\mathbf{x}_t} + \hat{\mathbf{a}}(1+\beta)\mathbf{p}_t + \hat{\mathbf{a}}\beta\big(\mathbf{S}_{g_1}(\mathbf{p}_{g_1}^k - \mathbf{u}_1^k) + \ldots + \mathbf{S}_{g_m}(\mathbf{p}_{g_m}^k - \mathbf{u}_m^k) + \mathbf{z}^k - \mathbf{v}^k\big)\Big)$$

where $\hat{\mathbf{a}} = \left((1+\beta)I + \beta\left(\mathbf{S}_{g_1} + \cdots + \mathbf{S}_{g_m}\right)\right)^{-1}$ and $\prod_{\Delta_n}$ is a projection to the probability simplex as in [13].

$\mathbf{p}_{g_i}$-updates: We solve for each $\mathbf{p}_{g_i}$ in parallel by projecting to the $L1$ ball of radius κ as in [13]

(4.21) $$\mathbf{p}_{g_i} = \prod_{\|\cdot\|_1 \le \kappa}\left(\mathbf{S}_{g_i}\mathbf{p}^{k+1} + \mathbf{u}_i^k\right)\ .$$

z-update: We solve for $\mathbf{z}$ by using the soft-thresholding operator $S_\rho(a)$ [20]

(4.22) $$\mathbf{z} = S_{\lambda/\beta}(\mathbf{p}^{k+1} - \mathbf{p}_t + \mathbf{v}^k)\ .$$

$\mathbf{u}_i$-updates: The $\mathbf{u}_i$-updates are already in closed form and can be computed in parallel for all i.

v-updates: The $\mathbf{v}$-update is already in closed form.

Algorithm 1 shows the complete ADMM based algorithm with the closed form updates. The stopping criteria for the ORASD algorithm is based on the primal and dual residuals from [4]. Algorithm 2 is our diversified online portfolio selection algorithm. It uses the ORASD algorithm to compute $\mathbf{p}_{t+1}$. It takes in an additional parameter γ which is a fixed percentage charged for the total amount of transaction every day. S_t^γ is the transaction cost-adjusted cumulative wealth gain at the end of t days.

4.3 Regret Bound We sequentially invest with the diverse portfolios $\mathbf{p}_1, \cdots, \mathbf{p}_T$ obtained from Algorithm 2 and on day t suffer a loss $f_t(\mathbf{p}_t) = \eta\ell_t + \lambda\|\mathbf{p}_t - \mathbf{p}_{t-1}\|_1$, where $\ell_t = -\log(\mathbf{p}_t^\top \mathbf{x}_t)$. 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, 6, 17] and follow [9] closely with minor modifications.

Theorem 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,*

(4.23)

$$\eta\sum_{t=1}^{T}\ell_t(\mathbf{p}_t)+\lambda\sum_{t=2}^{T}\|\mathbf{p}_t-\mathbf{p}_{t-1}\|_1-\eta\sum_{t=1}^{T}\ell_t(\mathbf{p}^*) \leq O(\sqrt{T}),$$

where ℓ_t is a strongly convex function and the sequence $\mathbf{p}_t$ and the fixed optimal portfolio $\mathbf{p}^*$ all lie in $\mathcal{P} := \{\mathbf{p} \,|\, \mathbf{p} \in \Delta_n, \|\mathbf{p}\|^{\mathcal{G}}_{(\infty,1)} \leq \kappa\}$.

5 Experiments and Results

5.1 Datasets The experiments were conducted on data taken from the New York Stock Exchange (NYSE) and the Standard & Poor's 500 (S&P 500) stock market index. The NYSE dataset [6] consists of 36 stocks with data accumulated over a period of 22 years from July 3, 1962 to December 31, 1984 and has been widely used in the literature [1, 3, 6, 17]. The S&P500 dataset [7] consists of 263 stocks which were present in the S&P500 index in 2010 and were alive since 1990.

5.2 Methodology and Parameter Setting In all our experiments we start with \$1 as our initial investment and an initial portfolio which is uniformly distributed over all the stocks. We use Algorithm 2 to obtain our portfolios sequentially and compute the transaction cost-adjusted wealth each day.

For our experiments, we utilize the follow method for computing the groups. For the previous num_{lag} days, we compute the correlation graph C over the stocks. To compute structurally related groups, we set a correlation threshold ϵ on the edges of the graph and if an edge has weight $< \epsilon$ then it is removed resulting in C_ϵ. For each stock in C_ϵ, we construct a group around it by including its neighbors within k hops away. For example, if $k = 1$ then we construct group g_i by including stock s_i and only its directly connected neighbors in the group. As such, there will be exactly n groups but the size of each group may vary. For our experiments, we allow the groups to change each day.

Since the two datasets are very different in nature (stock composition and duration), we experimented with various parameter values. We found stable behavior across the following range of parameters: $num_{lag} \in \{5, 10, 15\}$, $\epsilon = \{0.90, 0.95\}$, and $\beta = 2$. Additionally, we experimented extensively with a large range of values for η and λ from 10^{-6} to 10^3 and values for κ from 0.1 to 1.0 to observe their effect on our portfolio. Moreover, we chose a reasonable range of γ values between 0% and 2% to compute the proportional transaction costs incurred due to the portfolio update every day. We have illustrated some of our results with representative plots from either the NYSE or S&P500 dataset.

We use the wealth obtained from OLU [9], EG [17], a uniform constant rebalanced portfolio (U-CRP), and a buy-and-hold strategy as benchmarks for our experiments. For U-CRP, we make trades to rebalance the portfolio at the end of each day after the market movement has driven it away from the uniform distribution. For the buy-and-hold case we start with a uniformly distributed portfolio and do a hold on the positions thereafter, i.e., no trades. We do not consider algorithms such as Anticor [3] or OLMAR [22], which are good heuristics but without regret bounds.

5.3 Effect of the $L_{(\infty,1)}$ group norm and $\boldsymbol{\kappa}$ The $L_{(\infty,1)}$ group norm encourages diversity between the groups and sparsity within the groups. This structure is further effected by the value of κ which has direct control over the level of diversity. κ has an effect on (a) the value of $\|\mathbf{p}_t\|^{\mathcal{G}}_{(\infty,1)}$ each day, (b) the group weight $\|\mathbf{p}_{g_i}\|_1 \; \forall i$, and (c) the number of active groups.

(a) Value of $\|\mathbf{p}_t\|^{\mathcal{G}}_{(\infty,1)}$ With the diversity inducing constraint $\|\mathbf{p}_t\|^{\mathcal{G}}_{(\infty,1)} \leq \kappa$ we are encouraging different levels of diversity depending on the value of κ. From Figure 1, we can see the effect κ has on $\|\mathbf{p}_t\|^{\mathcal{G}}_{(\infty,1)}$ with $\eta = 100$ and $\lambda = 10^{-6}$. With a low κ value of 0.4, $\|\mathbf{p}_t\|^{\mathcal{G}}_{(\infty,1)}$ is small with many days seeing the value exactly equal to κ. As we increase κ, we see that the value moves along with κ with many days seeing the value equal κ. This behavior is consistent with aggressive trading due to the high η and low λ values. ORASD tries to invest as much wealth as allowed into groups that performed well in the past. With higher λ this behavior is not as aggressive and the value more slowly moves with κ and becomes more spread out.

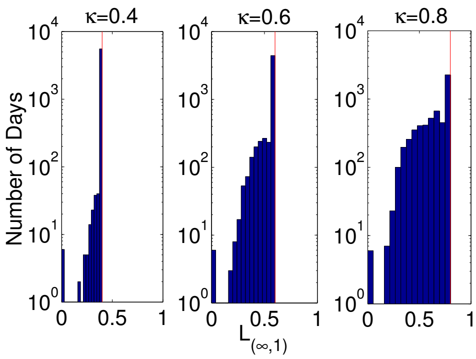


Figure 1: Total value of $\|\mathbf{p}_t\|^{\mathcal{G}}_{(\infty,1)}$ with vertical red lines marking the value of κ. With an aggressive trading strategy, the value closely follows the increasing κ.

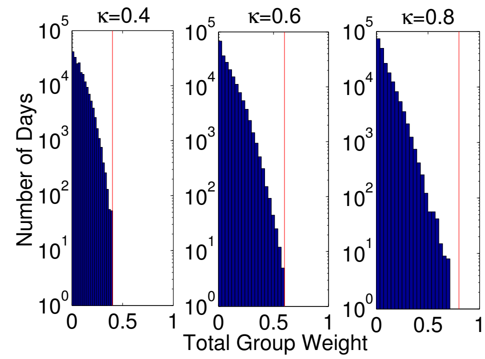


Figure 2: Total group weight $\|\mathbf{p}_{g_i}\|_1\ \forall i$ with vertical red lines marking the value of κ. With low κ we are forced to diversify but less so as κ increases.

(b) Group Weight $\|\mathbf{p}_{g_i}\|_1$ Not only is the value of $\|\mathbf{p}_t\|^{\mathcal{G}}_{(\infty,1)}$ affected by κ, but so are the group weights $\|\mathbf{p}_{g_i}\|_1\ \forall i$. Each of these are constrained to be less than or equal to κ and how close each group's weight gets to κ is of interest. If there are many days which see many groups with weight close to κ and many groups with little weight then this suggests ORASD is focusing on a few groups to invest the majority of wealth in at each day. If, however, there are many days with small group weight and few with large this suggests ORASD is investing a small amount of wealth in many groups and has a more diversified portfolio.

From Figure 2 we can see with $\eta = 10^{-3}$ and $\lambda = 10^{-6}$ ORASD utilizes a more conservative trading strategy. We can see this from how slowly the distribution moves with κ and from how many days see a small amount of wealth invested in each group. With low η and λ, ORASD does not trade aggressively and is further limited by κ. The portfolio is diversified with small κ and becomes slightly less so as we increase κ.

(c) Number of Active Groups We define an active group as a group which has a significant percent of wealth invested in it. The number of active groups can be a measure of how diverse a portfolio is. If there are many active groups this implies a diverse portfolio where few active groups does not. From Figure 3 we can see that for low $\kappa = 0.1$, the number of active groups is reasonably high with around 20 groups active out of 263 (about 8%). With low κ we are forcing a diverse portfolio therefore many groups have a significant amount of wealth invested in them. For high $\kappa = 0.9$, we see that the number of active groups drops to around 3 groups (about 1%). This implies that ORASD is focusing on a handful of well performing groups. This trading strategy has a potentially high reward but it also carries high risk. We can see how adjusting the value of κ can provide us the flexibility to use drastically different trading strategies.

5.4 Risk and κ Even though our online resource allocation with structured diversification framework (3.6) does not explicitly take risk into account, we can effectively control risk using κ as a proxy. We observe how the value of κ affects the amount of risk our diverse online portfolio is exposed to using three common measures of risk: (a) covariance, (b) Sharpe ratio, and (c) Sortino ratio.

(a) Covariance We compute the covariance $\mathbf{\Sigma}_t$ using the previous num_{lag} 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 $\alpha_{cov}{=}\mathbf{p}_t^\top\mathbf{\Sigma}_t\mathbf{p}_t/\mathbf{u}^\top\mathbf{\Sigma}_t\mathbf{u}$. High α_{cov} implies high risk and low α_{cov} implies low risk.

(b) Sharpe ratio The Sharpe ratio [25] measures

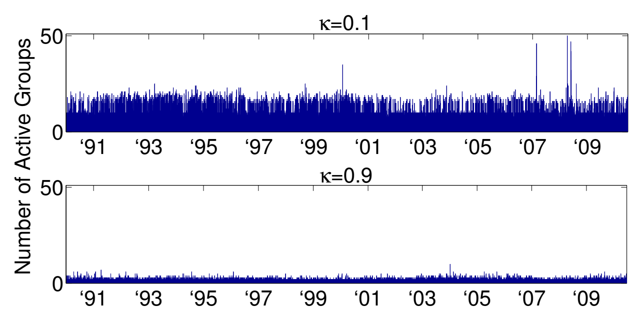
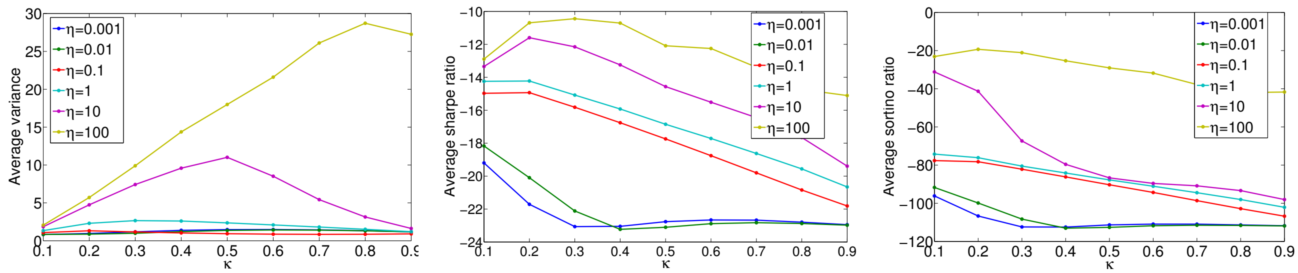


Figure 3: Number of active groups for the S&P500 dataset. For low κ we trade with a more conservative, diversified portfolio while with high κ we are more aggressive and risk seeking.



(a) Covariance risk, α_{cov} (b) Sharpe ratio, α_{sharpe} (c) Sortino ratio, $\alpha_{sortino}$

Figure 4: For α_{cov} with low η, the risk stays low for varying κ. For large η, the risk increases with an increasing κ. For both α_{sharpe} and $\alpha_{sortino}$, the risk-adjusted return is higher for higher η and deceases as we increase κ. From this figure, we can see that if we want to control the risk exposure, we can effectively do it by controlling κ.

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 $\alpha_{sharpe}=(R-R_b)/\sqrt{var(R-R_b)}$ 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 [26] 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 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).

From Figure 4, we can see the behavior of the three measures of risk for varying values of η and κ with $\lambda = 10^{-6}$. For α_{cov} in 4(a), with low values of η the risk is low and stays low as we increase κ. However, once we have higher η values, the risk starts to increase up to a point before it starts to show a decreasing trend as we increase κ. With higher η, the portfolio is focusing more on maximizing returns and is only investing in stocks that performed well in the past. This behavior does afford the portfolio the ability to earn huge amounts of wealth, however, as we can see it also exposes the investor to huge amounts of risk. In 4(b) and 4(c), we see that the risk-adjusted return increases as we increase η however, the curve trends downwards as we increase κ. This again implies that as κ increases so does the risk. From these plots we can see that κ is a good proxy to risk and we can therefore control risk by setting κ.

5.5 Transaction Cost-Adjusted Wealth To evaluate the practical application of ORASD we analyze the performance by calculating the transaction cost-adjusted cumulative wealth. We do this for varying values of κ to get a sense of the tradeoff between risk and return. We compare the performance of ORASD to the state-of-the-art algorithms OLU, EG, U-CRP, buy-and-hold, and the best stock in hindsight (without transaction costs) with empirically determined parameters.

From Figure 5 we can see that for optimal η and λ values and varying values of κ, ORASD returns more wealth than all the other competing algorithms for the NYSE dataset. ORASD earns \$163.07 compared to \$54.14 for the single best stock (Morris), \$50.80 for OLU, \$27.08 for U-CRP, \$26.70 for EG, and \$14.50 for buy-and-hold. ORASD returns over 3x as much as OLU and the best single stock we could have chosen

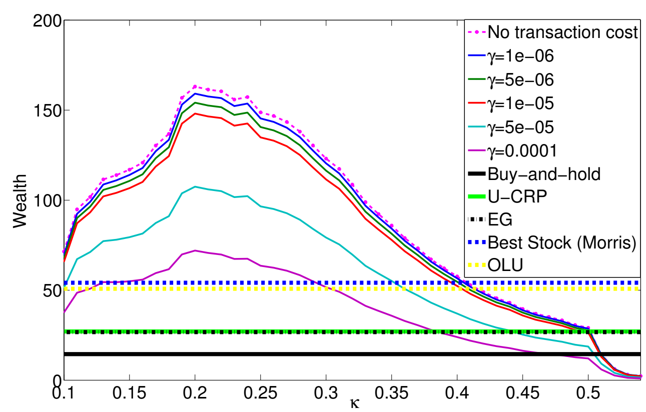
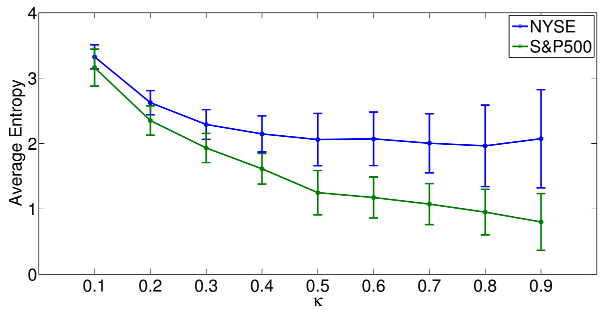
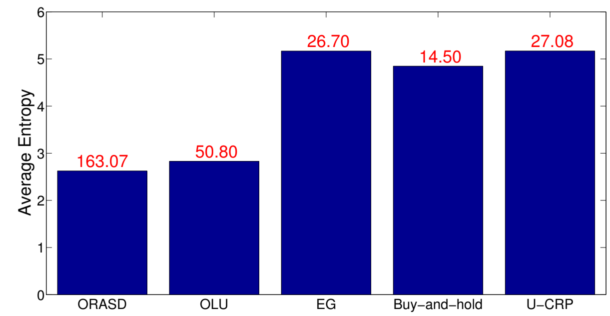


Figure 5: Transaction cost-adjusted wealth for the NYSE dataset. ORASD returns more wealth than competing algorithms even with transaction costs.



(a) Entropy of ORASD for η=100, λ=10^{-6}. High entropy implies the investment is more spread out. As κ increases, the entropy decreases which implies that the portfolio is concentrated on a single group of stocks.



(b) Entropy of ORASD and competing algorithms with cumulative wealth. ORASD has around the same level of diversification as OLU but returns more wealth.

Figure 6: Average entropy for ORASD and competing algorithms. We can see that as κ decreases so does the entropy. This indicates that the portfolio is becoming less diverse and as such may be exposed to more risk.

in hindsight. We can see that ORASD also returns more wealth than the competing algorithms even with transaction costs.

5.6 Diversification We saw from Figure 3 that one way to measure the diversity of a portfolio is by the number of active groups. Another measure commonly used is the entropy of the portfolio. Entropy measures how spread out a distribution is. If we have a portfolio with high entropy this implies that the portfolio is invested in many groups and is more diversified where with low entropy the portfolio only has investments in a few groups and is not as diversified.

From Figure 6(a), we can see that as κ increases the entropy decreases. This shows that with high κ the portfolio is focusing on a few groups to invest in and as such may be exposed to more risk. This again gives evidence that κ can be used as a proxy to risk.

Finally, we want to compare how diversified ORASD is against competing algorithms. We can see from Figure 6(b) that ORASD has the lowest entropy but is close to that of OLU. However, ORASD returns more than 3x as much wealth as OLU for the same level of diversity and risk exposure.

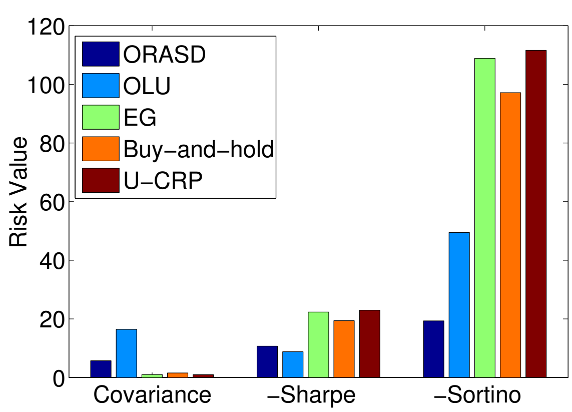


Figure 7: Risk comparison between competing algorithms for the NYSE dataset. We plot the negative Sharpe and Sortino ratios therefore the higher the bar the higher the risk.

5.7 Risk Comparison From Section 5.5, we saw that ORASD is able to return more wealth than all of the state-of-the-art algorithms. 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. To be consistent in the plot 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, a higher bar height implies higher risk.

In Figure 7 we can see that ORASD has lower α_{cov} than OLU but higher than the others with $\kappa = 0.2$, however, the other algorithms returned far less wealth. There is a balance between risk taken and potential returns. For α_{sharpe}, we can see that ORASD and OLU are reasonably close and both have smaller risk than the other algorithms. For $\alpha_{sortino}$, ORASD has the smallest risk. This can be explained by the fact that both the Sharpe and Sortino ratios compute the risk-adjusted return and the competing algorithms do not return much wealth so the compensation for the risk is low.

6 Conclusion

In this paper, we have developed an online resource allocation with structured diversification framework and an online learning algorithm (ORASD) and showed how it can be applied to the problem of online portfolio selection. Our analysis shows that ORASD is competitive with reasonable fixed strategies which have the power of hindsight. Our experimental results illustrate the effect of the diversity inducing group norm $L_{(\infty,1)}$ on the diversification of the portfolio, risk exposure, and wealth. We show that ORASD is able to outperform the state-of-the-art online portfolio selection algorithms in terms of wealth earned even with reasonable transaction costs and risk exposure. In the future we wish to explore extending our framework to allow long and short positions as well as meta portfolios.

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. AB also acknowledges support from IBM and Yahoo. The authors would also like to thank Maria Gini for her helpful feedback.

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] Maria-Florina Balcan and Avrim Blum. Mechanism design via machine learning. In *FOCS*, pages 605–614, Washington, DC, USA, 2005. IEEE Computer Society.

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

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

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

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

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

[8] Puja Das and Arindam Banerjee. Online quadratically constrained convex optimization with applications to risk adjusted portfolio selection. Technical report, Department of Computer Science and Engineering, University of Minnesota, 2012.

[9] Puja Das, Nicholas Johnson, and Arindam Banerjee. Online lazy updates for portfolio selection with transaction costs. In *AAAI*, pages 202–208, 2013.

[10] Puja Das, Nicholas Johnson, and Arindam Banerjee. Online portfolio selection with group sparsity. In *AAAI*, pages 1185–1191, 2014.

[11] Nikhil R. Devanur and Thomas P. Hayes. The adwords problem: Online keyword matching with budgeted bidders under random permutations. In *EC*, pages 71–78, New York, NY, USA, 2009. ACM.

[12] Dmitri A. Dolgov and Edmund H. Durfee. Resource allocation among agents with mdp-induced preferences. *JAIR*, pages 27–505, 2006.

[13] J. Duchi, Y. Singer, and T. Chandra. Efficient projections onto the l1-ball for learning in high dimensions. In *ICML*, 2008.

[14] Stefano Ermon, Jon Conrad, Carla Gomes, and Bart Selman. Risk-sensitive policies for sustainable renewable resource allocation. In *IJCAI*, pages 1942–1948. AAAI Press, 2011.

[15] Gagan Goel and Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. In *SODA*, pages 982–991, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[16] Daniel Golovin, Andreas Krause, Beth Gardner, Sarah J. Converse, and Steve Morey. Dynamic resource allocation in conservation planning. In *AAAI*, 2011.

[17] D. Helmbold, E. Schapire, Y. Singer, and M. Warmuth. On-line portfolio selection using multiplicative weights. *Mathematical Finance*, 8(4):325–347, 1998.

[18] J. W. Horwood. Risk-sensitive optimal harvesting and control of biological populations. *Mathematical Medicine and Biology*, 13(1):35–71, 1996.

[19] Ronald A. Howard and James E. Matheson. Risk-sensitive markov decision processes. *Management Science*, 18(7):pp. 356–369, 1972.

[20] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for hierarchical sparse coding. *JMLR*, pages 2297–2334, 2010.

[21] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for sparse hierarchical dictionary learning. In *ICML*, 2010.

[22] B. Li and S. Hoi. On-line portfolio selection with moving average reversion. In *ICML*, 2012.

[23] Julien Mairal, Rodolphe Jenatton, Francis R. Bach, and Guillaume R. Obozinski. Network flow algorithms for structured sparsity. In *NIPS*, pages 1558–1566. 2010.

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

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

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

[27] Shan Xue, Alan Fern, and Daniel Sheldon. Dynamic resource allocation for optimizing population diffusion. In *AISTATS*, pages 1033–1041, 2014.