Optimal Allocation Strategies for the Dark Pool Problem

Alekh Agarwal
University of California, Berkeley
alekh@cs.berkeley.edu

Peter Bartlett
University of California, Berkeley
bartlett@cs.berkeley.edu

Max Dama
University of California, Berkeley
maxdama@berkeley.edu

Abstract

We study the problem of allocating stocks to *dark pools*. We propose and analyze an optimal approach for allocations, if continuous-valued allocations are allowed. We also propose a modification for the case when only integer-valued allocations are possible. We extend the previous work on this problem (Ganchev et al., 2009) to adversarial scenarios, while also improving on their results in the iid setup. The resulting algorithms are efficient, and perform well in simulations under stochastic and adversarial inputs.

1 Introduction

In this paper we consider the problem of allocating stocks to *dark pools*. As described by Ganchev et al. (2009), dark pools are a recent type of stock exchange that are designed to facilitate large transactions. A key aspect of dark pools is the *censored feedback* that the trader receives. At every round the trader has a certain number V^t of shares to allocate amongst K different dark pools. The dark pool i trades as many of the allocated shares v_i as it can with the available liquidity. The trader only finds out how many of these allocated shares were successfully traded at each dark pool, but not how many would have been traded if more were allocated.

It is natural to assume that the actions of the trader affect the volume available at all dark pools at later times. Similarly, it seems natural that at a given time, the liquidities available at different venues should be correlated: we would expect counterparties to distribute large trades across many dark pools, simultaneously affecting their liquidity. Furthermore, in a realistic scenario, these variables are governed not only by the trader's actions, but also by the actions of other competing traders, each trying to maximize profits. Since the gain of one trader is at the expense of another, this problem naturally lends itself to an adversarial analysis. Generalizing the setup of Ganchev et al. (2009), we assume that the sequences of volumes and available liquidities are chosen by an adversary who knows the previous allocations of our algorithm.

We propose an exponentiated gradient (henceforth EG) style algorithm that has an optimal regret guarantee against the best allocation strategy in hindsight. Our algorithm uses a parametrization that allows it to handle the problem of changing constraint sets easily. Through a standard online to batch conversion, this also yields a significantly better algorithm in the iid setup studied in Ganchev et al. (2009). However, the EG algorithm has the drawback that it recommends continuous-valued allocations. We describe how the problem of allocating an integral number of shares closely resembles a multi-armed bandit problem. As a result, we use ideas from the Exp3 algorithm for adversarial bandit problems (Auer et al., 2003) to design an algorithm that produces integer-valued allocations and enjoys a regret of order $T^{2/3}$ with high probability. While this regret bound holds in an adversarial setting, it also implies an improvement on Ganchev et al. (2009) in an iid setting.

In the next section we will describe the problem setup in more detail and survey previous work. We will describe the EG algorithm for continuous allocations and prove its regret bound and optimality in Section 3. In Section 4 we describe the algorithm for integer valued allocations. Section 4.3 discusses implementation issues. Finally we present experiments comparing our algorithms with that of Ganchev et al. (2009) using the data simulator described in their paper.

2 Setup and Related Work

We generalize the setup of Ganchev et al. (2009). A learning algorithm receives a sequence of volumes $V^1, \ldots V^T$ where $V^t \in \{1, \ldots, V\}$. It has K available

Appearing in Proceedings of the 13^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9.

venues, amongst which it can allocate up to V^t units at time t. The learner chooses an allocation v_i^t for the i_{th} venue at time t that satisfies $\sum_{i=1}^K v_i^t \leq V^t$.

Each venue has a maximum consumption level s_i^t. The learner then receives the number of units $r_i^t = \min(v_i^t, s_i^t)$ consumed at venue i. We allow the sequence of volumes and maximum consumption levels to be chosen adversarially, i.e. V^t, s_i^t can depend on $\{v_i^1, \ldots, v_i^{t-1}\}_{i=1}^K$. We measure the performance of our learner in terms of its regret

$$R_T = \max \sum_{t=1}^{T} \sum_{i=1}^{K} \min(u_i^t, s_i^t) - \min(v_i^t, s_i^t)$$

where the outer maximization is over the vector opt $\in \{1, \ldots, K\}^V$ and

$$u_i^t = \sum_{v=1}^{V^t} \mathbb{I}(\text{opt}_v = i),$$

i.e., we compete against any strategy that chooses a fixed sequence of venues $\text{opt}_1, \ldots, \text{opt}_V$ and always allocates the vth unit to venue opt_v.

The works most closely related to ours are Ganchev et al. (2009) and Huh and Rusmevichientong (2009). In the first paper, the authors consider the sequence of volumes $V^1, \ldots, V^T$ and allocation limits s_i^t to be distributed in an iid fashion. They propose an algorithm based on Kaplan-Meier estimators. Their algorithm mimics an optimal allocation strategy by estimating the tail probabilities of s_i^t being larger than a given value. They show that the allocations of their algorithm are ϵ-suboptimal with probability at most $1 - \epsilon$ after seeing sufficiently many samples. Theorem 1 in Ganchev et al. (2009) shows that, if the s_i^t is chosen iid, then the optimal strategy always allocates the ith unit to a fixed venue. This justifies our definition of regret in comparison to this class of strategies. In Huh and Rusmevichientong (2009) the authors consider a stochastic gradient desscent algorithm for 1 venue when the demands are drawn in an i.i.d. fasion. They also discuss integral allocations through rounding, but assume side information to get different rates of convergence than us.

The ideas used in our paper draw on the rich literature on online adversarial learning. The algorithm of Section 3 is based on the classical EG algorithm (Littlestone and Warmuth, 1994). When playing integral allocations, we describe how the multi-armed bandits problem is a special case of our problem for $V = 1$. For the general case, we describe an adaptation of the Exp3 algorithm (Auer et al., 2003) for adversarial multi-armed bandits. To provide regret bounds that hold with high probability, we use a variance correction similar to the Exp3.P algorithm (Auer et al., 2003). Our lower bounds use techniques similar to lower bound arguments for experts prediction and multi-armed bandits. The efficient implementation of our algorithm relies on greedy approximation techniques in Hilbert spaces.

3 Optimal algorithm for fractional allocations

Although the dark pools problem requires us to allocate an integral number of shares at every venue, we start by studying the simpler case where we can allocate any positive value for every venue, so long as they satisfy $\sum_{i=1}^K v_i^t \leq V^t$. We note that the reward function $r_i^t = \min(v_i^t, s_i^t)$ is concave in allocations v_i^t.

Maximization of concave functions is well understood, even in an adversarial scenario through approaches such as online gradient ascent. We note that in this problem, the algorithm has access to the subgradient of the reward function. To see this, we define

$$g_i^t = \begin{cases} 1 & \text{if } r_i^t = v_i^t \\ 0 & \text{if } r_i^t < v_i^t \end{cases} \tag{1}$$

Then it is easy to check that g_i^t can be constructed from the feedback we receive, and it lies in the subgradient set $\frac{\partial r_i^t}{\partial v_i^t}$. Hence, we can run a standard online (sub)gradient ascent algorithm on this sequence of reward functions. However, the allocations v_i^t are chosen from a different set $S_t = \{\vec{v}^t : \sum_{i=1}^K v_i^t \leq V^t\}$ at every round. Using standard online gradient ascent analysis, we can demonstrate a low regret only against a comparator that lies in the intersection of all these constraint sets $\cap_{t=1}^T S_t$. However the regret guarantee can be rather meaningless if V^t is extremely small at even a single round. Ideally, we would like to compete with an optimal allocation strategy like Ganchev et al. (2009). A slightly different parameterization allows us to do exactly that.

Let us define $\Delta_K^V = \{(x^1, \ldots, x^V) : \sum_{i=1}^K x_i^v = 1 \; \forall v \leq V\}$ to be the Cartesian product of V simplices, each in $\mathbb{R}^K$. Then we can construct an algorithm for allocations as follows: for each unit $v = \{1, \ldots, V\}$, we have a distribution over the venues $\{1, \ldots, K\}$ where that unit is allocated. At time t, the algorithm plays $v_i^t = \sum_{v=1}^{V^t} x_{t,i}^v$. It is clear that this allocation satisfies the volume constraint.

The comparator is now defined as a fixed point $u \in \Delta_K^V$. We compete with the strategy that plays according to $v_i^t = \sum_{v=1}^{V^t} u_i^v$. Then the best comparator u is equivalent to the best fixed allocation strategy opt $\in \{1, \ldots, K\}^V$. It is also clear that if we can com-

pete with the best strategy in an adversarial setup, online to batch conversion techniques (see Cesa-Bianchi et al. (2001)) will give a small expected error in the case where the volumes and maximum consumptions are drawn in an iid fashion.

3.1 Algorithm and upper bound

An online exponentiated gradient ascent algorithm for this setup is presented in Algorithm 1.

Algorithm 1 Exponentiated gradient algorithm for continuous-valued allocations to dark pools

Input learning rate η, bound on volumes V.
Initialize $x^v_{1,i} = \frac{1}{K}$ for $v \in \{1,\dots,V\}$, $i \in \{1,\dots,K\}$.
for $t = 1,\dots,T$ **do**
  Set $v^t_i = \sum_{v=1}^{V^t} x^v_{t,i}$.
  Receive $r^t_i = \min\{v^t_i, s^t_i\}$.
  Set g^t_i as defined in Equation (1).
  Set $g^v_{t,i} = g^t_i$ if $v \le V^t$, 0 otherwise.
  Update $x^v_{t+1,i} \propto x^v_{t,i}\exp(\eta g^v_{t,i})$.
end for

It can be shown that the algorithm enjoys the following regret guaranteee.

Theorem 1. *For any choices of the volumes $V^t \in [0,V]$ and of the maximum consumption levels s^t_i, the regret of Algorithm 1 with $\eta = \sqrt{\frac{\ln K}{(e-2)T}}$ over T rounds is $O(V\sqrt{T\ln K})$.*

Proof. The regret is defined as

$$R_T = \max_{u\in\Delta_K^V} \sum_{t=1}^T\sum_{i=1}^K \min\left(\sum_{v=1}^{V^t} u^v_i, s^t_i\right) - \sum_{t=1}^T\sum_{i=1}^K \min\left(v^t_i, s^t_i\right)$$
$$\le \sum_{t=1}^T\sum_{v=1}^{V^t} (u^v - x^v_t)^\top g^v_t.$$

Following the proof of Theorem 11.3 from Cesa-Bianchi and Lugosi (2006), we define $\nu^v_i = \eta g^v_{t,i} - \eta (g^v_t)^\top x^v_t$. Also, we note that the gradient is zero for $v > V^t$. So we can sum over v from 1 to V rather than V^t. Then we bound the regret as

$$\sum_{t=1}^T\sum_{v=1}^V \left[(u^v - x^v_t)^\top g^v_t - \frac{1}{\eta}\ln\left(\sum_{i=1}^K x^v_{t,i}\exp(\nu^v_i)\right) + \frac{1}{\eta}\ln\left(\sum_{i=1}^K x^v_{t,i}\exp(\nu^v_i)\right)\right].$$

Some rewriting and simplification gives the bound

$$\frac{1}{\eta}\sum_{t=1}^T\sum_{v=1}^V\left[\sum_{i=1}^K u^v_i \ln\left(\frac{\exp\left(\eta g^v_{t,i}\right)}{\sum_{i=1}^K \exp\left(\eta g^v_{t,i}\right)}\right) + \ln\left(\sum_{i=1}^K x^v_{t,i} e^{\nu^v_i}\right)\right]$$
$$= \frac{1}{\eta}\sum_{t=1}^T\sum_{v=1}^V\left[u^v_i \ln\left(\frac{x^v_{t+1,i}}{x^v_{t,i}}\right) + \ln\left(\sum_{i=1}^K x^v_{t,i}\exp(\nu^v_i)\right)\right]$$
$$\le \frac{1}{\eta}\sum_{v=1}^V\left[\mathrm{KL}(u^v||x^v_1) + \sum_{t=1}^T \ln\left(\sum_{i=1}^K x^v_{t,i}\exp(\nu^v_i)\right)\right].$$

Here, the last line uses the definition of KL-divergence and the fact that the telescoping terms cancel out. Now $g^v_{t,i} \le 1$ so that $\nu^v_i \le \eta$. If $\eta \le 1$, then it is easy to verify that $\exp(\nu^v_i) \le 1 + \nu^v_i + (e-2)\left(\nu^v_i\right)^2$. We also note that $\sum_{i=1}^K x^v_{t,i}\nu^v_i = 0$.

Also, each of the KL divergence terms in the above display is equal to $\ln K$. This is because the optimal comparator will have a 1 for exactly one venue for each unit v. As we choose x^v_1 to be uniform over all venues, we get the KL divergence between a vertex of the K-simplex and the uniform distribution which, is $\ln K$.

Hence we bound the regret as

$$\frac{1}{\eta}V\ln K + \frac{1}{\eta}\sum_{t=1}^T\sum_{v=1}^V \ln\left(\sum_{i=1}^K x^v_{t,i}\left(1 + \nu^v_i + (e-2)\left(\nu^v_i\right)^2\right)\right)$$
$$\le \frac{1}{\eta}V\ln K + \frac{1}{\eta}\sum_{t=1}^T\sum_{v=1}^V (e-2)\eta^2$$
$$= \frac{1}{\eta}V\ln K + (e-2)\eta VT \le 3V\sqrt{T\ln K},$$

where the last step follows from setting $\eta = \sqrt{\frac{\ln K}{(e-2)T}}$. □

3.2 Lower bound and minimax optimality

We will now show that the online exponentiated gradient ascent algorithm in Algorithm 1 has the best regret guarantee possible. We start by noting that a a regret bound of $O(\sqrt{T\ln K})$ is known to be optimal for the experts prediction problem (Haussler et al., 1998; Abernethy et al., 2009). Hence we can show the optimality of our algorithm for $V = 1$ by reducing experts prediction problem to the dark pools problem. Recall that in the experts prediction problem, the algorithm picks an expert from $1,\dots,K$ according to a probability distribution p_t at round t. Then it receives a vector of rewards ρ_t with $\rho_{t,i} \in [0,1]$, $i = 1,\dots,K$. In order to describe a reduction, we need to map the allocations of an algorithm for the dark pools problem to the probabilities for experts, and map the rewards of experts to the liquidities at each venue.

We consider a special setting where $V_t = 1$ at all times. Since $V_t = 1$, the allocations of any dark pools algorithm are probabilities– they are non-negative and add

to 1. Hence we set $p_{t,i} = v_i^t$. We also set the liquidity $s_i^t = \rho_{t,i} p_{t,i}$. Then the net reward of a dark pools algorithm at round t is:

$$\sum_{i=1}^{K} \min(s_i^t, v_i^t) = \sum_{i=1}^{K} \min(\rho_{t,i} p_{t,i}, p_{t,i}) = \sum_{i=1}^{K} \rho_{t,i} p_{t,i},$$

where the last line follows from the observation that $0 \leq \rho_{t,i} \leq 1$. Hence the net reward of the dark pools problem is same as the expected reward in the experts prediction problem. Using the known lower bounds on the optimal regret in experts prediction problems, we get:

$$\begin{aligned}
&\max_{s^1,\ldots,s^T} \max_{u \in \Delta_K} \sum_{t=1}^{T} \sum_{i=1}^{K} \left[\min\left(u_i, s_i^t\right) - \min(v_i^t, s_i^t)\right] \\
&= \max_{\rho_1,\ldots,\rho_T} \max_{i} \sum_{t=1}^{T} \left[\rho_{t,i} - \sum_{j=1}^{K} \rho_{t,j} p_{t,j}\right] \\
&= \Omega(\sqrt{T \ln K}).
\end{aligned}$$

We also note that the regret in the experts prediction problem scales linearly with the scaling of the rewards. Hence, if the rewards take values in $[0, V]$, then the worst case regret of any algorithm is guaranteed to be $\Omega(V\sqrt{T \ln K})$.

For arbitrary V, we let V^t identically equal to V. We would now like to reduce the experts prediction problem where every expert's reward is a value in $[0, V]$. At every round, we receive a vector of allocations v_i^t and set $p_{t,i} = v_i^t / V$. We receive the rewards $\rho_{t,i}$ from the experts problem, and assign the liquidities $s_i^t = \rho_{t,i} p_{t,i} \in [0, V]$. Furthermore,

$$\min(s_i^t, v_i^t) = V \min\left(\frac{s_i^t}{V}, p_{t,i}\right) = \rho_{t,i} p_{t,i}.$$

The last step relies on observing that $\rho_{t,i} \leq V$ so that $\rho_{t,i} p_{t,i} / V \leq p_{t,i}$. Now we can argue that the regrets of the two problems are identical as before. Hence the optimal regret in the dark pools problem is at least $\Omega(V\sqrt{T \ln K})$. As Algorithm 1 gets the same bound up to constant factors in a harder adversarial setting than used in the lower bounds, we conclude that it attains the minimax optimal regret up to constant factors.

4 Algorithm for integral allocations

While the above algorithm is simple and optimal in theory, it is a bit unrealistic as it can recommend we allocate 1.5 units to a venue, for example. One might choose to naively round the recommendations of the algorithm, but such a rounding would incur an additional approximation error which in general could be as large as $O(T)$. In this section we describe a low regret algorithm that allocates an integral number of units to each venue.

To get some intuition about an algorithm for this scenario, consider the case when $V = 1$. Then the algorithm has to allocate 1 unit to a venue at every round. It receives feedback about the maximum allocation level s_i^t only at the venue where $v_i^t = 1$. This is clearly a reformulation of the classical K-armed bandits problem. An adaptation of Algorithm 1 that uses the Exp3 algorithm (Auer et al., 2003) would hence attain a regret bound of $O(\sqrt{TK \ln K})$ for $V = 1$. Contrasting this with the bound of Theorem 1 for $V = 1$, we can easily see that the regret for playing integral allocations can be higher than that of continuous allocations by a factor of up to $\sqrt{K}$. Indeed we will now show a modification of the Exp3 approach that works for arbitrary values of V. We will also show a lower bound. The upper bound shows that our algorithm incurs $O(T^{2/3})$ regret in expectation, which does not match the $\Omega(\sqrt{T})$ lower bound. However, it is still a significant improvement on Ganchev et al. (2009) as we will discusss later.

4.1 Algorithm and upper bound

We need some new notation before describing the algorithm. For a fractional allocation v_i^t, we let $f_i^t = \lfloor v_i^t \rfloor$ and $d_i^t = v_i^t - \lfloor v_i^t \rfloor$.

Now suppose we have a strategy that wants to allocate v_i^t units to venue i at time t. Suppose that we instead allocate $u_i^t = f_i^t$ units with probability $1 - d_i^t$ and $u_i^t = f_i^t + 1$ units with probability d_i^t. Using the fact that the maximum consumption limits are integral too

$$\begin{aligned}
\mathbb{E}\min(u_i^t, s_i^t) &= d_i^t \min(f_i^t + 1, s_i^t) + (1 - d_i^t)\min(f_i^t, s_i^t) \\
&= \begin{cases} s_i^t & \text{if } s_i^t \leq f_i^t \\ f_i^t + d_i^i & \text{if } s_i^t \geq f_i^t + 1 \end{cases} \\
&= \min(v_i^t, s_i^t).
\end{aligned}$$

Thus, playing an integral allocation u_i^t according to such a scheme would be unbiased in expectation. Of course we need to ensure that we don't violate the constraint $\sum_{i=1}^{K} u_i^t \leq V^t$ in this process. To do so, we let $\sum_{i=1}^{K} d_i^t = V_t - \sum_{i=1}^{K} f_i^t = m$. Then we will use a distribution over subsets of $\{1, \ldots, K\}$ of size m that has the property that i_{th} element gets sampled with probability d_i^t. For all the elements in the sampled subset, we set $u_i^t = f_i^t + 1$. It is clear that if there is such a distribution, then we will have the unbiasedness needed above. It will also ensure feasibility of u_i^t if v_i^t was a feasible allocation. Our next result shows that such a distribution always exists.

Theorem 2. *Let $0 \leq d_i^t < 1$, $\sum_{i=1}^{K} d_i^t = m$ for $m \geq 1$. Then there is always a distribution over subsets of $\{1, \ldots, K\}$ of size m such that the i_{th} element is sampled with probability d_i^t.*

Proof. Proof is by induction on K. For the case

$K = 2, m = 1$, we sample the first element with probability d_1^t. If it is not picked, we pick element 2. It is clear that the marginals are correct establishing the base case. Let us assume the claim holds up to $K-1$ for all $m \leq K-1$. Consider the inductive step for some K, m. We are given a set of marginals, $0 \leq d_i^t < 1$, $\sum_{i=1}^K d_i^t = m$. We would like a distribution p on subsets of size m of $\{1, \ldots, K\}$ that matches these marginals. We partition these subsets into two groups; those that do and do not contain the first element. We correspondingly partition $p = (p_1, p_2)$. Let $N_1 = \binom{K-1}{m-1}$ and $N_2 = \binom{K-1}{m}$ be the number of subsets in the two cases. Then we want $\sum_{i=1}^{N_1} p(i) = \sum_{i=1}^{N_1} p_1(i) = d_1^t$ in order to get the right marginal at element 1. Hence, we can write $p_1 = d_1^t q_1$, $p_2 = (1 - d_1^t)q_2$ for some distributions q_1 and q_2 on N_1 and N_2 subsets respectively. Now we write

$$d_i^t = \left(\frac{(m-1)d_1^t}{m - d_1^t} + \frac{m(1-d_1^t)}{m - d_1^t} \right) d_i^t \tag{2}$$

for $i > 1$. Then

$$\sum_{i=2}^K \frac{(m-1)}{m - d_1^t} d_i^t = m - 1, \quad \sum_{i=2}^K \frac{m}{m - d_1^t} d_i^t = m \tag{3}$$

are marginals on subsets of size $m-1$ and m respectively of $\{1, \ldots, K-1\}$, and are in $[0,1]$ as $\sum_{i=2}^K d_i^t = m - d_1^t$. Hence there exist distributions q_1 and q_2 that attain these marginals using the inductive hypothesis. We set $p_1 = d_1^t q_1$, $p_2 = (1 - d_1^t)q_2$. Then Equations 2 and 3 together imply that we get the correct marginals for every element. □

For any allocation sequence v^t, let $p(d^t)$ be the probability distribution over subsets of $\{1, \ldots, K\}$ guaranteed by Theorem 2. For some constant $\gamma \in (0, 1]$, let $\bar{d}_{t,i} = (1-\gamma)d_i^t + \frac{\gamma m}{K}$. Then let $p(\bar{d}_{t,i})$ be a distribution over subsets that samples the i_{th} venue with probability $\bar{d}_{t,i}$. We can construct this by mixing $p(d_i^t)$ which exists by Theorem 2 with a uniform distribution over subsets of size m. Also, we let $\tilde{V}_{t,i} \leq V_t$ be the largest index v_0 such that $\sum_{v=1}^{v_0} x_{t,i}^v \leq f_i^t$. We define a gradient estimator:

$$\tilde{g}_{t,i}^v = \begin{cases} \mathbb{I}(s_i^t \geq f_i^t) - \frac{\mathbb{I}(s_i^t = f_i^t)\mathbb{I}(u_i^t = \lceil v_i^t \rceil)}{\bar{d}_{t,i}} \text{ if } v \leq \tilde{V}_{t,i} \\ \frac{\mathbb{I}(s_i^t \geq v_i^t)\mathbb{I}(u_i^t = \lceil v_i^t \rceil)}{\bar{d}_{t,i}} \text{ if } \tilde{V}_{t,i} + 1 \leq v \leq V^t. \end{cases} \tag{4}$$

To see why this gradient estimator is good, we first note that the gradient of the objective function at v_i^t can be written as

$$g_{t,i}^v = \mathbb{I}(s_i^t \geq v_i^t) = \mathbb{I}(s_i^t \geq f_i^t) - \mathbb{I}(s_i^t = f_i^t),$$

when $v \leq V^t$. Then we can easily show the following useful lemma.

Lemma 1. *If an algorithm plays $u_i^t = \lceil v_i^t \rceil$ with probability $\bar{d}_{t,i}$ and $u_i^t = f_i^t$ otherwise, then $\tilde{g}_t$ as described in Equation (4) is an unbiased estimator of the gradient at $(v_1^t, \ldots, v_K^t)$.*

An algorithm for playing integer-valued allocations at every round is shown in Algorithm 2.

Algorithm 2 An algorithm for playing integer-valued allocations to the dark pools

Input learning rate η, threshold γ, bound on volumes V.
Initialize $x_{1,i}^v = \frac{1}{K}$ for $v = \{1, \ldots, V\}$.
for $t = 1 \ldots T$ **do**
  Set $v_i^t = \sum_{v=1}^{V^t} x_{t,i}^v$.
  Let $p(\bar{d}_{t,i})$ be the distribution over subsets from Theorem 2.
  Sample a subset of size $m = \sum_{i=1}^K \bar{d}_{t,i}$ according to $p(\bar{d}_{t,i})$.
  Play $u_i^t = f_i^t + 1$ if i is in the subset sampled, $u_i^t = f_i^t$ otherwise.
  Receive $r_i^t = \min(u_i^t, s_i^t)$.
  Set $\tilde{g}_{t,i}^v$ as defined in Equation (4).
  Update $x_{t+1,i}^v \propto x_{t,i}^v \exp(\eta \tilde{g}_{t,i}^v)$.
end for

We can also demonstrate a guarantee on the expected regret of this algorithm.

Theorem 3. *Algorithm 2, with $\eta = \left(\frac{V(\ln K)^2}{KT^2}\right)^{1/3}$, has expected regret over T rounds of $O((VTK)^{2/3}(\ln K)^{1/3})$, where V is the bound on volumes V^t, and the volumes and maximum consumption levels s_i^t are chosen by an oblivious adversary.*

An oblivious adversary is one that chooses V^t and s_i^t without seeing the algorithm's (random) allocations u_i^t. We note that the requirement that the adversary is oblivious can be removed by proving a high probability bound. In the full version (Agarwal et al., 2009), we describe a modification of Algorithm 2 that enjoys such a guarantee.

Proof. Since the adversary is oblivious, we can fix a comparator $u \in \Delta_K^V$ ahead of time. For the remainder, we let $\mathbb{E}_t$ denote conditional expectation at time t conditioned on the past moves of algorithm and adversary. Then the expected regret is

$$\mathbb{E}\left[\sum_{t=1}^T \sum_{i=1}^K \min\left(\sum_{v=1}^V u_i^v, s_i^t\right) - \sum_{t=1}^T \sum_{i=1}^K \min\left(u_i^t, s_i^t\right)\right]$$
$$\leq \mathbb{E}\left[\sum_{t=1}^T \sum_{i=1}^K \min\left(\sum_{v=1}^V u_i^v, s_i^t\right) - \sum_{t=1}^T \sum_{i=1}^K \min(v_i^t, s_i^t)\right] + \gamma TK.$$

Here, the second step follows from the fact that u_i^t would be unbiased for v_i^t without for the $\frac{\gamma m}{K}$ adjustment. However, this adjustment costs us at most $\gamma \sum_{t=1}^T m_t \leq \gamma TK$ in terms of expected regret over T rounds. The first term is as if we had played the continuous valued allocation v_i^t itself. Again using the

concavity of our reward function

$$R_T(u) \le \mathbb{E}\left[\sum_{v=1}^{V}(u^v - x_t^v)^\top g_t^v\right] + \gamma TK$$
$$= \mathbb{E}\left[\sum_{v=1}^{V}(u^v - x_t^v)^\top (\mathbb{E}_t \tilde{g}_t^v)\right] + \gamma TK.$$

Here the last step follows from noting that $\tilde{g}_t$ is unbiased estimator of g_t by construction just like in Exp3 (Auer et al., 2003). Now we note that the algorithm is doing exponentiated gradient descent on the sequence $\tilde{g}_t$. Hence, we can proceed as in the proof of Theorem 1 to obtain

$$R_T(u) \le \frac{1}{\eta} V \ln K + \frac{1}{\eta}\mathbb{E}\sum_{t=1}^{T}\sum_{v=1}^{V}\ln\left(\sum_{i=1}^{K} x_{t,i}^v \exp(\nu_i^v)\right) + \gamma TK,$$

where $\nu_i^v = \eta\tilde{g}_{t,i}^v - \eta(\tilde{g}_t^v)^\top x_t^v$ as before. Assuming a choice of η such that $\eta\tilde{g}_{t,i}^v \le 1$, we note again that $\nu_i^v \le 1$. So we can use the quadratic bound on exponential again and simplify as before to get

$$R_T(u) \le \frac{1}{\eta} V \ln K + \frac{1}{\eta}\mathbb{E}\sum_{t=1}^{T}\sum_{v=1}^{V}\sum_{i=1}^{K} x_{t,i}^v(\nu_i^v)^2 + \gamma TK$$
$$= \frac{1}{\eta} V \ln K + \eta\mathbb{E}\sum_{t=1}^{T}\sum_{v=1}^{V}\sum_{i=1}^{K} x_{t,i}^v(\tilde{g}_{t,i}^v)^2 + \gamma TK.$$

Now we can swap the sum over V and i to obtain

$$R_T(u) \le \frac{1}{\eta} V \ln K + \eta\mathbb{E}\sum_{t=1}^{T}\sum_{i=1}^{K}\sum_{v=1}^{V} x_{t,i}^v(\tilde{g}_{t,i}^v)^2 + \gamma TK$$
$$= \frac{1}{\eta} V \ln K + \eta\mathbb{E}\sum_{t=1}^{T}\sum_{i=1}^{K}\left[\sum_{v=1}^{\tilde{V}_{t,i}} x_{t,i}^v(\tilde{g}_{t,i}^v)^2 + \sum_{v=\tilde{V}_{t,i}+1}^{V^t} x_{t,i}^v(\tilde{g}_{t,i}^v)^2\right] + \gamma TK.$$

Now we look at the two gradient terms separately.

$$\mathbb{E}_t\sum_{v=1}^{\tilde{V}_{t,i}} x_{t,i}^v(\tilde{g}_{t,i}^v)^2 = \sum_{v=1}^{\tilde{V}_{t,i}} x_{t,i}^v\left\{\bar{d}_{t,i}\left(\mathbb{I}(s_i^t \ge f_i^t) - \frac{\mathbb{I}(s_i^t = f_i^t)}{\bar{d}_{t,i}}\right)^2 + (1-\bar{d}_{t,i})\mathbb{I}(s_i^t \ge v_i^t)\right\}$$
$$\le 2v_t^i + 2v_t^i\frac{K}{\gamma}.$$

Here, we used the fact that $\bar{d}_{t,i} \ge \frac{\gamma}{K}$ as $m \ge 1$ and indicator variables are bounded by 1. Hence

$$\mathbb{E}\sum_{t=1}^{T}\sum_{i=1}^{K}\sum_{v=1}^{\tilde{V}_{t,i}} x_{t,i}^v(\tilde{g}_{t,i}^v)^2 \le 2TV + 2\frac{TVK}{\gamma}$$

using $\sum_{i=1}^{T} v_i^t \le V$. Next we examine the second gradient term

$$\mathbb{E}_t\sum_{v=\tilde{V}_{t,i}+1}^{V^t} x_{t,i}^v(\tilde{g}_{t,i}^v)^2 = \mathbb{E}_t\sum_{v=\tilde{V}_{t,i}+1}^{V^t} x_{t,i}^v(\tilde{g}_{t,i}^{V^t})^2$$
$$= \mathbb{E}_t d_i^t(\tilde{g}_{t,i}^{V^t})^2 \le \bar{d}_{t,i}d_i^t\frac{1}{(\bar{d}_{t,i})^2} \quad \le 2 \text{ if } \gamma \le \tfrac{1}{2}.$$

Hence, $\mathbb{E}\sum_{t=1}^{T}\sum_{i=1}^{K}\sum_{v=\tilde{V}_{t,i}+1}^{V^t} x_{t,i}^v(\tilde{g}_{t,i}^v)^2 \le 2TK$. Substituting the above terms in the bound, we get

$$R_t(u) \le \frac{1}{\eta} V \ln K + 2\eta\left(TV + \frac{TVK}{\gamma} + TK\right) + \gamma TK.$$

Optimizing for η, γ gives

$$R_T(u) \le 6(VTK)^{2/3}(\ln K)^{1/3}.$$

□

We note that the term responsible for $O(T^{2/3})$ regret is $\frac{\mathbb{I}(s_i^t = f_i^t)}{\bar{d}_{t,i}}$. While we assume that this can accumulate at every round in the worst case, it seems unlikely that the liquidity s_i^t will be equal to f_i^t very frequently. In particular, if the s_i^t's are generated by a stochastic process, one can control this probability using the distribution of s_i^t and obtain improved regret bounds.

By using standard variance correction techniques (Auer et al., 2003), (Abernethy and Rakhlin, 2009), we can show a similar bound with high probability. Combining it with a union bound over all comparators allows us to extend the results of Theorem 3 to adaptive adversaries too. We omit these standard steps for lack of space, and details can be found in the full version of the paper (Agarwal et al., 2009).

Comparison with results of Ganchev et al. (2009): We note that although our results are in the adversarial setup, the same results also apply to iid problems. In particular, using online-to-batch conversion techniques (Cesa-Bianchi et al., 2001), we can show that, after T rounds, with high probability the allocations of our algorithm on each round is within $\widetilde{O}(V^2T^{-1/3}K^{2/3})$ of the optimal allocation. This is a significant improvement on the result of Ganchev et al. (2009): it is straightforward to check that the proof they provide gives a corresponding upper bound no better than $O(T^{-1/4})$. As we shall see, the generalization to adversarial setups leads to improved performance in simulations.

4.2 Lower bound on regret

As mentioned in the previous section, the problem of K-armed bandits is a special case of the dark pools problem with integral allocations. Hence, we would like to leverage the proof techniques from existing lower bounds on the optimal regret in the K-armed bandits problem. As before we consider a special case with $V_t = V$ at every round. Following Auer et al. (2003), we construct K different distributions for generating the liquidities s_i^t. At each round, the i_{th} distribution samples $s_i^t = V$ with probability $\left(\frac{1}{2} + \epsilon\right)$ and $s_j^i = V$ with probability $\frac{1}{2}$ for $j \ne i$. We now mimic the proof of Theorem 5.1 in Auer et al. (2003). We can arrive at the following result.

Theorem 4. *Any algorithm that plays integer valued allocations has expected regret that is* $\Omega\left(\sqrt{TV(K+V\ln K)}\right)$.

Proof. Using arguments similar to Auer et al. (2003), it is easy to show that the expected regret is lower bounded by $\Omega(\sqrt{TVK})$. We also note that the lower bound of $\Omega(V\sqrt{T\ln K})$ shown for continuous-valued allocations applies to the integer-valued case as well. Combining the two, we get that the regret is

$$\Omega(\max\{\sqrt{TVK}, V\sqrt{T\ln K}\}) = \Omega\left(\sqrt{T}\left(\sqrt{VK}+V\sqrt{\ln K}\right)\right).$$

□

There is a gap between our lower and upper bounds in this case. We do not know which bound is loose.

4.3 Efficient sampling for integral allocations

All that remains to specify in Algorithm 2 is the construction of the distribution p over subsets at every round. Since we don't know what the distribution is, it would seem that we cannot sample from it easily. If K is small, one can use non-negative least squares to find the distribution that has the given marginals. However, once the number of venues K is large, p is a distribution over $\binom{K}{m}$ subsets, for which the least squares solver might be too slow. One way around is to use the idea of greedy approximations in Hilbert Spaces. We can greedily construct a distribution on subsets which matches the marginals on every element approximately in an efficient manner. Exact sampling from the distribution without ever constructing it explicitly is also possible. The explicit algorithms giving the implementations can be found in the full version of the paper (Agarwal et al., 2009).

5 Experimental results

We compared four methods experimentally. We refer to Algorithms 1 and 2 as ExpGrad and Exp3 respectively. We also run the Optimistic Kaplan Meier estimator based algorithm of Ganchev et al. (2009), which is called OptKM. Finally we implemented the parametric maximum likelihood estimation-allocation based algorithm described in (Ganchev et al., 2009) as well, which we call ParML. As we did not have access to real dark pool data, we decided to implement a data simulator similar to Ganchev et al. (2009). We used a combination of a *Zero Bin* parameter and power law distribution to generate the s_i^t's while the sequence V^t was kept fixed. Parameters for the Zero Bin and power law were set to lie in the same regimes as the ones observed in the real data of Ganchev et al. (2009).

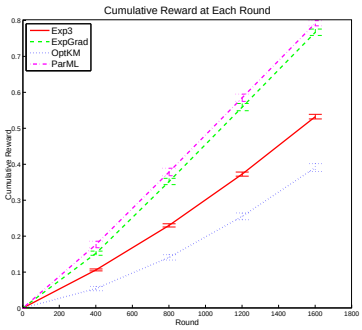


Figure 1: *Cumulative rewards for each algorithm as a function of the number of rounds when run on the parametric model of (Ganchev et al., 2009) averaged over 100 trials*

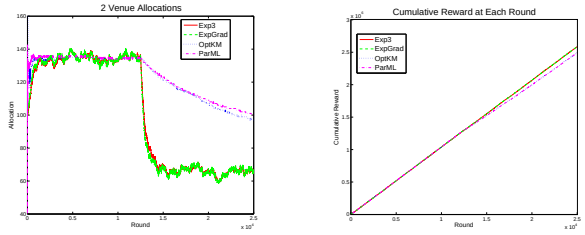
We started by generating the data from the parametric model of Ganchev et al. (2009). We used 48 venues, $T = 2000$ to match their experiments. The values of s_t^i's were sampled iid from Zero Bin+Power law distributions with appropriately chosen parameters. A plot of the resulting cumulative rewards averaged over 100 trial runs is in in Figure 1.

We see that ParML has a slightly superior performance on this data, understandably as the data is being generated from the specific parametric model that the algorithm is designed for. However, ExpGrad gets net allocations quite close to ParML. Furthermore, both Exp3 and ExpGrad are far superior to the performance of OptKM which is our true competitor in some sense being a non-parametric approach just like ours.

Next, we study the performance of all four algorithms under a variety of adversarial scenarios. We start with a simple setup of two venues. The parameters of the power law initially favor Venue 1 for 12500 rounds, and then we switch the power law parameters to favor Venue 2. We study both the cumulative rewards as well as the allocations to both venues for each algorithm. Clearly an algorithm will be more robust to adversarial perturbations if it can detect this change quickly and switch its allocations accordingly. We show the results of this experiment in Figure 2.

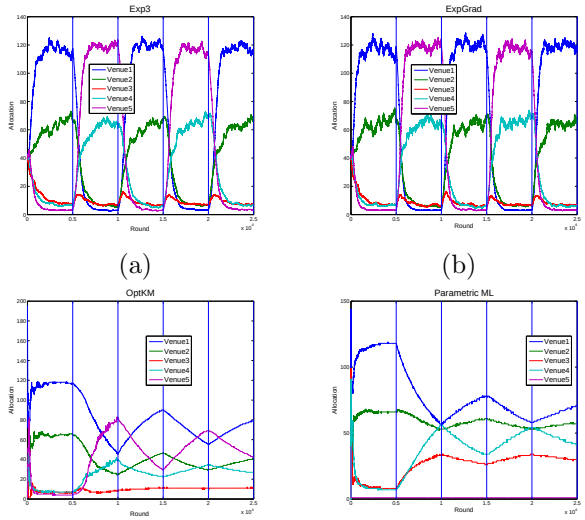
Because of just 2 venues, rounding has a rather negligible effect in this case and both our methods have an almost identical performance. Our algorithms ExpGrad and Exp3 switch much faster to the new optimal venue when distributions switch. Consequently, the cumulative reward of both our algorithms also turns out significantly higher as shown in Figure 2(b).

We wanted to investigate how this behavior changes when the switching involves a larger number of venues. We created another experiment where there are 5 venues, maximum volume $V = 200$. Venues 1 and 5 oscillate between getting very favorable and unfavorable



(a) (b)

Figure 2: *Allocations to the 2 venues and cumulative rewards for the different algorithms. Note the inability of* ParML *and* OptKM *to effectively switch between venues when distributions switch.* ExpGrad *and* Exp3 *also achieve higher cumulative rewards.*



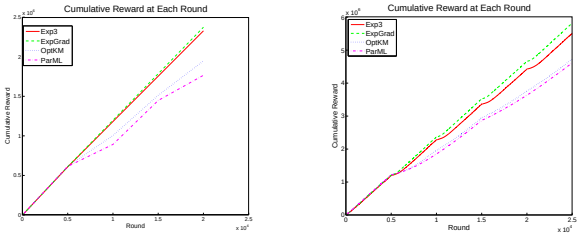
(a) (b) (c) (d)

Figure 3: *Allocations to the 5 venues for the different algorithms. Note the poor switching of* OptKM *between venues when distributions switch.* ParML *completely fails on this problem.* Exp3 *and* ExpGrad *correctly identify both long and short range trends (see text).*

values of the law exponent. Other venues also switch, but between less extreme values. Allocations to all 5 venues for each algorithm are shown in Figure 3.

Once again both Exp3 and ExpGrad identify both the long range trend (favorability of venues 1, 5 over the others) and short range trend (favoring venue 1 over 5 in certain phases). There is a gap between Exp3and ExpGrad this time, however, as rounding does start to play a role with 5 venues. OptKM adapts somewhat, although it still doesn't reach as high an allocation level as Exp3 after switching to a new venue. ParML adapts quite poorly to this switching as well. We also studied the behavior of algorithms as V is scaled on the same problem. Figure 4 plots the cumulative reward of each algorithm for $V = 200$ and $V = 400$. It is clear that ExpGrad and Exp3 still comprehensively outperform others.

In summary, it seems that our algorithms are competitive with those of Ganchev et al. (2009) when the data



(a) (b)

Figure 4: *Cumulative rewards for each algorithm when distributions switch between 5 venues, for $V = 200$ (left) and $V = 400$. Note the superior performance of* ExpGrad *and* Exp3.

is drawn from their parametric model. When their assumptions about iid data are not satisfied, we significantly outperform those algorithms. We note that we have only experimented with oblivious adversaries here. The gulf in performance may be even wider for adaptive adversaries.

Acknowledgements: We would like to thank Jennifer Wortman Vaughan and Warren Schudy for their helpful comments and suggestions. We gratefully acknowledge the support of NSF grants 0707060 and 0830410. Alekh is partially supported by a MSR Graduate fellowship.

References

Abernethy, J., Agarwal, A., Bartlett, P. L., and Rakhlin, A. (2009). A stochastic view of optimal regret through minimax duality. In *Proceedings of the 22nd Annual Conference on Learning Theory*.

Abernethy, J. and Rakhlin, A. (2009). Beating the adaptive bandit with high probability. In *Proceedings of COLT 2009*.

Agarwal, A., Bartlett, P. L., and Dama, M. (2009). Optimal allocation strategies for the dark pool problem. *CoRR, arXiv preprint*, abs/1003.2245.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2003). The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77.

Cesa-Bianchi, N., Conconi, A., and Gentile, C. (2001). On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50:2050–57.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, Learning and Games*. Cambridge University Press.

Ganchev, K., Kearns, M., Nevmyvaka, Y., and Vaughan, J. W. (2009). Censored exploration and the dark pool problem. In *Proceedings of Uncertainity in Artificial Intelligence, UAI 2009*.

Haussler, D., Kivinen, J., and Warmuth, M. K. (1998). Sequential prediction of individual sequences under general loss functions. *IEEE Transactions on Information Theory*, 44(5):1906–1925.

Huh, W. T. and Rusmevichientong, P. (2009). A nonparametric asymptotic analysis of inventory planning with censored demand. *Math. Oper. Res.*, 34(1):103–123.

Littlestone, N. and Warmuth, M. K. (1994). The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261.