An Online Allocation Problem: Dark Pools

Peter Bartlett
Statistics and EECS
UC Berkeley

Joint work with Alekh Agarwal and Max Dama.

slides at http://www.stat.berkeley.edu/∼bartlett

Prediction in Probabilistic Settings

- i.i.d. $(X, Y), (X_1, Y_1), \ldots, (X_n, Y_n)$ from $\mathcal{X} \times \mathcal{Y}$ (e.g., Y is preference score vector).
- Use data $(X_1, Y_1), \ldots, (X_n, Y_n)$ to choose $f_n : \mathcal{X} \to \mathcal{A}$ with small risk,

$$R(f_n) = \mathbf{E}\ell(Y, f_n(X)).$$

Online Learning

- Repeated game:

  Player chooses a_t

  Adversary reveals ℓ_t

- Example: $\ell_t(a_t) = \text{loss}(y_t, a_t(x_t))$.
- Aim: minimize $\sum_t \ell_t(a_t)$, compared to the best (in retrospect) from some class:

$$\text{regret} = \sum_t \ell_t(a_t) - \min_{a \in \mathcal{A}} \sum_t \ell_t(a).$$

- Data can be adversarially chosen.

Online Learning: Motivations

1. Adversarial model is appropriate for
   - Computer security.
   - Computational finance.
2. Understanding statistical prediction methods.
3. Online algorithms are also effective in probabilistic settings.

The Dark Pools Problem

- Computational finance: adversarial setting is appropriate.
- Online algorithm improves on best known algorithm for probabilistic setting.

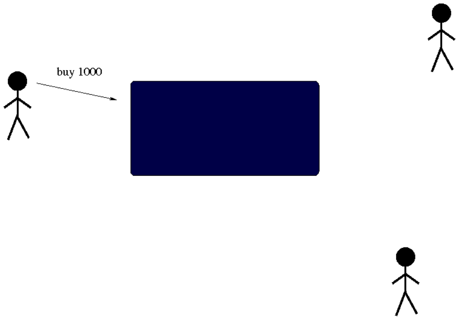
Dark Pools

Instinet,
Chi-X,
Knight Match, ...

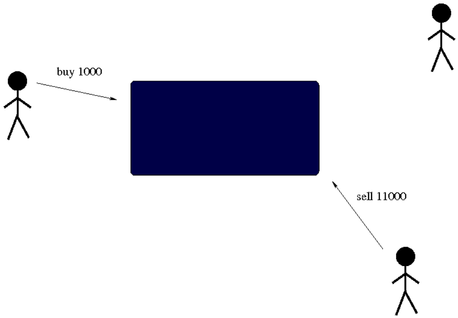
International Securities Exchange,
Investment Technology Group (POSIT),

- Crossing networks.
- Alternative to open exchanges.
- Avoid market impact by hiding transaction size and traders' identities.

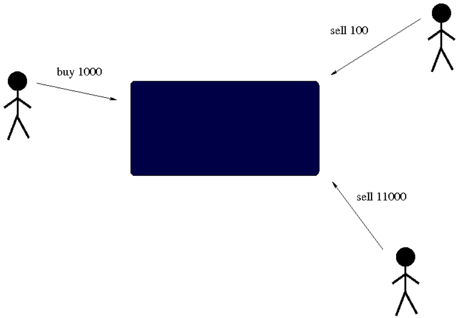
Dark Pools



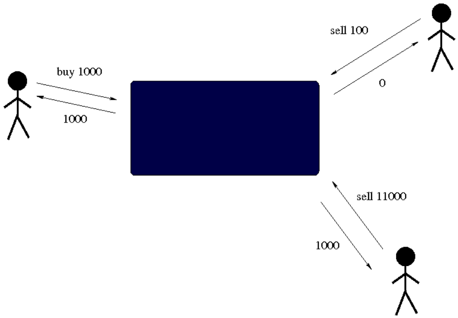
Dark Pools



Dark Pools



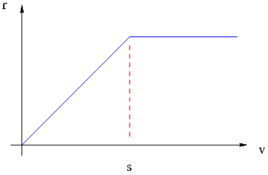
Dark Pools



Allocations for Dark Pools

The problem: Allocate orders to several dark pools so as to maximize the volume of transactions.

- Volume V^t must be allocated across K venues: $v_1^t, \ldots, v_K^t$, such that $\sum_{k=1}^{K} v_k^t = V^t$.
- Venue k can accommodate up to s_k^t, transacts $r_k^t = \min(v_k^t, s_k^t)$.



- The aim is to maximize $\displaystyle\sum_{t=1}^{T}\sum_{k=1}^{K} r_k^t$.

Allocations for Dark Pools

- Allocation $v_1^t, \ldots, v_K^t$ ranks the K venues.
- Loss is not discrete: it is summed across venues, and depends on the allocations in a piecewise-linear, convex, monotone way.

Allocations for Dark Pools: Probabilistic Assumptions

Previous work: (Ganchev, Kearns, Nevmyvaka and Wortman, 2008)

- Assume venue volumes are i.i.d.: $\{s_k^t,\ k = 1, \ldots, K,\ t = 1, \ldots, T\}$.
- In deciding how to allocate the first unit, choose the venue k where $Pr(s_k^t > 0)$ is largest.
- Allocate the second and subsequent units in decreasing order of venue tail probabilities.
- Algorithm: estimate the tail probabilities (Kaplan-Meier estimator—data is censored), and allocate as if the estimates are correct.

Allocations for Dark Pools: Adversarial Assumptions

I.i.d. is questionable:

- one party's gain is another's loss
- volume available now affects volume remaining in future
- volume available at one venue affects volume available at others

In the adversarial setting, we allow an arbitrary sequence of venue capacities (s_k^t), and of total volume to be allocated (V^t). The aim is to compete with any fixed allocation order.

Continuous Allocations

We wish to maximize a sum of (unknown) concave functions of the allocations:

$$J(v) = \sum_{t=1}^{T} \sum_{k=1}^{K} \min(v_k^t, s_k^t),$$

subject to the constraint $\sum_{k=1}^{K} v_k^t \leq V^t$.
The allocations are parameterized as distributions over the K venues:

$$x_t^1, x_t^2, \ldots \in \Delta_{K-1} = (K-1)\text{-simplex}.$$

Here, x_t^1 determines how the first unit is allocated, x_t^2 the second, ...

The algorithm allocates to the kth venue: $v_k^t = \sum_{v=1}^{V^t} x_{t,k}^v$.

Continuous Allocations

We wish to maximize a sum of (unknown) concave functions of the distributions:

$$J = \sum_{t=1}^{T} \sum_{k=1}^{K} \min(v_k^t(x_{t,k}^v), s_k^t).$$

Want small regret with respect to an arbitrary distribution x^v, and hence w.r.t. an arbitrary allocation.

$$\text{regret} = \sum_{t=1}^{T} \sum_{k=1}^{K} \min(v_k^t(x_k^v), s_k^t) - J.$$

Continuous Allocations

We use an exponentiated gradient algorithm:

Initialize $x^v_{1,i} = \frac{1}{K}$ for $v = \{1, \ldots, V\}$.
for $t = 1, \ldots, T$ **do**
  Set $v^t_k = \sum_{v=1}^{V^T} x^v_{t,k}$.
  Receive $r^t_k = \min\{v^t_k, s^t_k\}$.
  Set $g^v_{t,k} = \nabla_{x^v_{t,k}} J$.
  Update $x^v_{t+1,k} \propto x^v_{t,k} \exp(\eta g^v_{t,k})$.
end for

Continuous Allocations

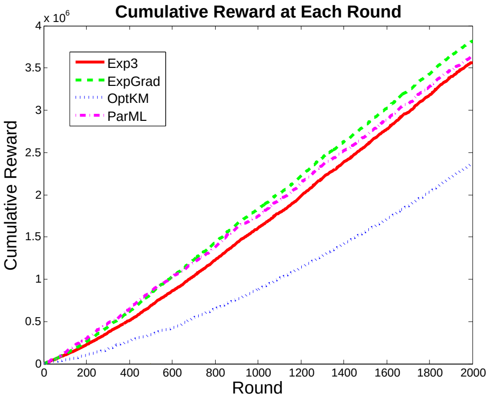
Theorem: For all choices of $V^t \leq V$ and of s_k^t, ExpGrad has regret no more than $3V\sqrt{T \ln K}$.

Continuous Allocations

Theorem: For all choices of $V^t \leq V$ and of s_k^t, ExpGrad has regret no more than $3V\sqrt{T \ln K}$.

Theorem: For every algorithm, there are sequences V^t and s_k^t such that regret is at least $V\sqrt{T \ln K}/16$.

Experimental results



Continuous Allocations: i.i.d. data

- Simple online-to-batch conversions show ExpGrad obtains per-trial utility within $O(T^{-1/2})$ of optimal.
- Ganchev et al bounds:
  per-trial utility within $O(T^{-1/4})$ of optimal.

Discrete allocations

- Trades occur in quantized parcels.
- Hence, we cannot allocate arbitrary values.
- This is analogous to a multi-arm bandit problem:
  - We cannot directly obtain the gradient at the current x.
  - But, we can estimate it using importance sampling ideas.

Theorem: There is an algorithm for discrete allocation with expected regret $\tilde{O}((VTK)^{2/3})$.
Any algorithm has regret $\tilde{\Omega}((VTK)^{1/2})$.

Dark Pools

- Allow adversarial choice of volumes and transactions.
- Per trial regret rate superior to previous best known bounds for probabilistic setting.
- In simulations, performance comparable to (correct) parametric model's, and superior to nonparametric estimate.