Prediction and sequential decision problems in adversarial environments

Peter Bartlett

Computer Science and Statistics
University of California at Berkeley

CDAR Symposium
October 16, 2015
Formulating decision problems as sequential games
Formulating decision problems as sequential games

- Decision problems: regression, classification, order allocation, dynamic pricing, portfolio optimization, option pricing.
Game-Theoretic Statistics

Formulating decision problems as sequential games

- Decision problems: regression, classification, order allocation, dynamic pricing, portfolio optimization, option pricing.
- Rather than model the process generating the data probabilistically, we view it as an adversary.
Formulating decision problems as sequential games

- Decision problems: regression, classification, order allocation, dynamic pricing, portfolio optimization, option pricing.
- Rather than model the process generating the data probabilistically, we view it as an adversary.

Decision-making = hedging against the future choices of the process generating the data.
**Outline**

- Decision problems as sequential games
  1. Allocation to dark pools
  2. Pricing options
  3. Linear regression
Outline

- Decision problems as sequential games
  1. Allocation to dark pools
  2. Pricing options
  3. Linear regression
Prediction as a game

A repeated game:

At round $t$:

1. Player chooses prediction $a_t \in A$.
2. Adversary chooses outcome $y_t \in Y$.
3. Player incurs loss $\ell(a_t, y_t)$.

Player's aim: Minimize regret:

$$T \sum_{t=1}^{T} \ell(a_t, y_t) - \min_{a \in A} T \sum_{t=1}^{T} \ell(a, y_t).$$

$$\ell(a_t, y_t) = \|a_t - y_t\|_2.$$
Prediction as a game

A repeated game:
At round $t$:

1. Player chooses prediction $a_t \in A$. 

Player incurs loss $\ell(a_t, y_t)$.

Player's aim: Minimize regret:

$$\sum_{t=1}^{T} \ell(a_t, y_t) - \min_{a \in A} \sum_{t=1}^{T} \ell(a, y_t).$$

$$\ell(a_t, y_t) = \|a_t - y_t\|_2.$$
Prediction as a game

A repeated game:

At round $t$:

1. Player chooses prediction $a_t \in \mathcal{A}$.
2. Adversary chooses outcome $y_t \in \mathcal{Y}$.
Prediction as a game

A repeated game:

At round $t$:

1. Player chooses prediction $a_t \in A$.
2. Adversary chooses outcome $y_t \in Y$.
3. Player incurs loss $\ell(a_t, y_t)$.

\[
\ell(a_t, y_t) = \|a_t - y_t\|^2.
\]
Prediction as a game

A repeated game:

At round $t$:

1. Player chooses prediction $a_t \in \mathcal{A}$.
2. Adversary chooses outcome $y_t \in \mathcal{Y}$.
3. Player incurs loss $\ell(a_t, y_t)$.

Player's aim:

Minimize regret:

$$\sum_{t=1}^{T} \ell(a_t, y_t) - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} \ell(a, y_t).$$
Prediction as a game

A repeated game:

At round $t$:

1. Player chooses prediction $a_t \in A$.
2. Adversary chooses outcome $y_t \in Y$.
3. Player incurs loss $\ell(a_t, y_t)$. 
Prediction as a game

A repeated game:

At round $t$:

1. Player chooses prediction $a_t \in \mathcal{A}$.
2. Adversary chooses outcome $y_t \in \mathcal{Y}$.
3. Player incurs loss $\ell(a_t, y_t)$.
Prediction as a game

A repeated game:

At round $t$:

1. Player chooses prediction $a_t \in A$.
2. Adversary chooses outcome $y_t \in Y$.
3. Player incurs loss $\ell(a_t, y_t)$. 

Player’s aim: 

Minimize regret:

$$
\sum_{t=1}^{T} \ell(a_t, y_t) - \min_{a \in A} \sum_{t=1}^{T} \ell(a, y_t).
$$

$$
\ell(a_t, y_t) = \|a_t - y_t\|_2.
$$
Prediction as a game

A repeated game:

At round \( t \):

1. Player chooses prediction \( a_t \in \mathcal{A} \).
2. Adversary chooses outcome \( y_t \in \mathcal{Y} \).
3. Player incurs loss \( \ell(a_t, y_t) \).
Prediction as a game

A repeated game:

At round $t$:

1. Player chooses prediction $a_t \in A$.
2. Adversary chooses outcome $y_t \in \mathcal{Y}$.
3. Player incurs loss $\ell(a_t, y_t)$. 

Player's aim: Minimize regret:

$$\sum_{t=1}^{T} \ell(a_t, y_t) - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} \ell(a, y_t).$$
Prediction as a game

A repeated game:

At round \( t \):
1. Player chooses prediction \( a_t \in \mathcal{A} \).
2. Adversary chooses outcome \( y_t \in \mathcal{Y} \).
3. Player incurs loss \( \ell(a_t, y_t) \).

Player’s aim:
Minimize regret:

\[
\sum_{t=1}^{T} \ell(a_t, y_t) - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} \ell(a, y_t).
\]
Online Prediction Games

The value of the game:

Minimax

Regret

\[ V_T(Y, A) = \min_{a_1 \in A} \max_{y_1 \in Y} \cdots \min_{a_T \in A} \max_{y_T \in Y} \left( \sum_{t=1}^{T} \ell(a_t, y_t) - \min_{a \in A} \sum_{t=1}^{T} \ell(a, y_t) \right) \]
Minimax Regret

\[
\left( \sum_{t=1}^{T} \ell(a_t, y_t) - \min_{a \in A} \sum_{t=1}^{T} \ell(a, y_t) \right)
\]
Online Prediction Games

Minimax Regret

$$\min_{a_1 \in \mathcal{A}} \left( \sum_{t=1}^{T} \ell(a_t, y_t) - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} \ell(a, y_t) \right)$$
Online Prediction Games

The value of the game:

\[
\min_{a_1 \in A} \max_{y_1 \in Y} \left( \sum_{t=1}^{T} \ell(a_t, y_t) - \min_{a \in A} \sum_{t=1}^{T} \ell(a, y_t) \right)
\]
Online Prediction Games

Minimax Regret

\[
\min_{a_1 \in \mathcal{A}} \max_{y_1 \in \mathcal{Y}} \cdots \min_{a_T \in \mathcal{A}} \left( \sum_{t=1}^{T} \ell(a_t, y_t) - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} \ell(a, y_t) \right)
\]
The value of the game:

\[ V(T, \mathcal{A}) = \min_{a_1 \in \mathcal{A}} \max_{y_1 \in \mathcal{Y}} \cdots \min_{a_T \in \mathcal{A}} \max_{y_T \in \mathcal{Y}} \left( \sum_{t=1}^{T} \ell(a_t, y_t) - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} \ell(a, y_t) \right) \]
Online Prediction Games

The value of the game: Minimax Regret

\[ V_T(\mathcal{Y}, \mathcal{A}) = \min_{a_1 \in \mathcal{A}} \max_{y_1 \in \mathcal{Y}} \cdots \min_{a_T \in \mathcal{A}} \max_{y_T \in \mathcal{Y}} \left( \sum_{t=1}^{T} \ell(a_t, y_t) - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} \ell(a, y_t) \right) \]
Online Prediction Games

The value of the game: Minimax Regret

\[ V_T(\mathcal{Y}, \mathcal{A}) = \min_{a_1 \in \mathcal{A}} \max_{y_1 \in \mathcal{Y}} \cdots \min_{a_T \in \mathcal{A}} \max_{y_T \in \mathcal{Y}} \left( \sum_{t=1}^{T} \ell(a_t, y_t) - \min_{a \in \mathcal{A}} \sum_{t=1}^{T} \ell(a, y_t) \right) \]
Online Prediction

<table>
<thead>
<tr>
<th>$a_t$</th>
<th>$y_t$</th>
<th>$\ell(a_t, y_t)$</th>
</tr>
</thead>
</table>

Density $\theta$ outcome $y_t - \log p_\theta(y_t)$

Regression $f_\theta(x_t)$ outcome $y_t - \|f_\theta(x_t) - y_t\|_2^2$

Bandit $p_t$ on $\{1, \ldots, k\}$ rewards $y \in \mathbb{R}^k - E I_t \sim p_t y_I_t$ (observe only $y_I_t$)
### Online Prediction

<table>
<thead>
<tr>
<th>Examples</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density estimation</td>
<td>$\ell(a_t, y_t) = -\log p_\theta(y_t)$</td>
</tr>
<tr>
<td></td>
<td>$a_t$</td>
</tr>
<tr>
<td></td>
<td>density $p_\theta$</td>
</tr>
</tbody>
</table>
## Online Prediction

### Examples

<table>
<thead>
<tr>
<th></th>
<th>$a_t$</th>
<th>$y_t$</th>
<th>$\ell(a_t, y_t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density estimation</td>
<td>density $p_\theta$</td>
<td>outcome $y_t$</td>
<td>$-\log p_\theta(y_t)$</td>
</tr>
<tr>
<td>Regression</td>
<td>$f_\theta(x_t)$</td>
<td>outcome $y_t$</td>
<td>$|f_\theta(x_t) - y_t|^2$</td>
</tr>
</tbody>
</table>
### Online Prediction

#### Examples

<table>
<thead>
<tr>
<th></th>
<th>$a_t$</th>
<th>$y_t$</th>
<th>$\ell(a_t, y_t)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density estimation</td>
<td>density $p_\theta$</td>
<td>outcome $y_t$</td>
<td>$- \log p_\theta(y_t)$</td>
</tr>
<tr>
<td>Regression</td>
<td>$f_\theta(x_t)$</td>
<td>outcome $y_t$</td>
<td>$|f_\theta(x_t) - y_t|^2$</td>
</tr>
<tr>
<td>Bandit</td>
<td>$p_t$ on ${1, \ldots, k}$</td>
<td>rewards $y \in \mathbb{R}^k$ (observe only $y_{I_t}$)</td>
<td>$-\mathbb{E}<em>{I_t \sim p_t} y</em>{I_t}$</td>
</tr>
</tbody>
</table>
Online Prediction

Probabilistic Model
- Batch
- Independent random data.
- Aim for small expected loss subsequently.

Adversarial Model
- Online
- Sequence of interactions with an adversary.
- Aim for small cumulative loss throughout.
Game-Theoretic Statistics

Why?

Weak assumptions on data
Streaming: appropriate for big data
Often no harder than the probabilistic formulation
Insight into robustness to probabilistic assumptions
Why?

- Weak assumptions on data
Game-Theoretic Statistics

Why?

- Weak assumptions on data
Game-Theoretic Statistics

Why?

- Weak assumptions on data
- Streaming: appropriate for big data
<table>
<thead>
<tr>
<th>Why?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weak assumptions on data</td>
</tr>
<tr>
<td>Streaming: appropriate for big data</td>
</tr>
<tr>
<td>Often no harder than the probabilistic formulation</td>
</tr>
</tbody>
</table>
Why?

- Weak assumptions on data
- Streaming: appropriate for big data
- Often no harder than the probabilistic formulation
- Insight into robustness to probabilistic assumptions
Online algorithms are also effective in probabilistic settings.

- Easy to convert an online algorithm to a batch algorithm.
- Easy to show that good online performance implies good i.i.d. performance, for example.
• Decision problems as sequential games
  1 Allocation to dark pools
  2 Pricing options
  3 Linear regression
Dark Pools Allocation

Joint work with Alekh Agarwal and Max Dama.

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

Instinet
BATS

Liquidnet
Investment Technology Group (POSIT)
Dark Pools Allocation

Joint work with Alekh Agarwal and Max Dama.

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

Instinet
BATS
Liquidnet
Investment Technology Group (POSIT)
Dark Pools

...
Dark Pools
Allocations for Dark Pools

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

At time $t$:
Allocations for Dark Pools

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

1. See the required volume $V^t$ to be allocated.
Allocations for Dark Pools

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

At time $t$:

1. See the required volume $V^t$ to be allocated.
2. Choose allocations $v_1^t, \ldots, v^K_t$ across the $K$ venues, such that $\sum_{k=1}^K v_k^t = V^t$. 

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

At time $t$:

1. See the required volume $V^t$ to be allocated.
2. Choose allocations $v^t_1, \ldots, v^t_K$ across the $K$ venues, such that
   \[ \sum_{k=1}^{K} v^t_k = V^t. \]
3. Venue $k$ can accommodate up to $s^t_k$, transacts $r^t_k = \min(v^t_k, s^t_k)$. 

\[ \text{The aim is to maximize } T \sum_{t=1}^{T} \sum_{k=1}^{K} r^t_k. \]
Allocations for Dark Pools

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

At time $t$:

1. See the required volume $V^t$ to be allocated.
2. Choose allocations $v_1^t, \ldots, v_K^t$ across the $K$ venues, such that $\sum_{k=1}^{K} v_k^t = V^t$.
3. 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 $\sum_{t=1}^{T} \sum_{k=1}^{K} r_k^t$. 

\[ \sum_{t=1}^{T} \sum_{k=1}^{K} r_k^t \]
Previous approaches: probabilistic. (Ganchev, Kearns, Nevmyvaka and Wortman, 2008)
Previous approaches: probabilistic. (Ganchev, Kearns, Nevmyvaka and Wortman, 2008)

- Assume independent venue volumes:
  \[ \{s^t_k, k = 1, \ldots, K, t = 1, \ldots, T\}. \]
Allocations for Dark Pools: Probabilistic Assumptions

Previous approaches: probabilistic. (Ganchev, Kearns, Nevmyvaka and Wortman, 2008)

- Assume independent venue volumes:
  \( \{s^t_k, \ k = 1, \ldots, K, \ t = 1, \ldots, T\} \).

- In deciding how to allocate the first unit, choose the venue \( k \) where \( Pr(s^t_k > 0) \) is largest.
Previous approaches: probabilistic. (Ganchev, Kearns, Nevmyvaka and Wortman, 2008)

- Assume independent venue volumes:
  \[ s_k^t, \ k = 1, \ldots, K, \ t = 1, \ldots, T \].

- 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.
Previous approaches: probabilistic. (Ganchev, Kearns, Nevmyvaka and Wortman, 2008)

- Assume independent venue volumes: 
  \[ \{s^t_k, \ k = 1, \ldots, K, \ t = 1, \ldots, T\} \].

- In deciding how to allocate the first unit, choose the venue \( k \) where \( Pr(s^t_k > 0) \) 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

Independence assumption 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
Independence assumption 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 \((V^t)\).
Continuous Allocations: Concave maximization

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

\[ J(v) = \sum_{t=1}^{T} \sum_{k=1}^{K} \min(v^t_k, s^t_k), \]

subject to the constraint \( \sum_{k=1}^{K} v^t_k \leq V^t \).
Continuous Allocations: Concave maximization

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

\[ J(v) = \sum_{t=1}^{T} \sum_{k=1}^{K} \min(v_{t,k}^t, s_{k}^t), \]

subject to the constraint \( \sum_{k=1}^{K} v_{t,k}^t \leq V^t \).

The allocations are parameterized as distributions over the \( K \) venues:

\[ x_1^t, x_2^t, \ldots \in \Delta_{K-1} = (K - 1)\text{-simplex}. \]

Here, \( x_1^t \) determines how the first unit is allocated, \( x_2^t \) the second, ...

Allocate to the \( k \)th venue:

\[ v_{k}^t = \sum_{v=1}^{V^t} x_{t,v,k}^v. \]
We wish to maximize a sum of (unknown) concave functions of the distributions:

\[ J = \sum_{t=1}^{T} \sum_{k=1}^{K} \min(v^t_k(x^v_{t,k}), s^t_k). \]
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: Online Convex Optimization

Exponentiated gradient algorithm

- Mirror descent (each step optimizes a sum of a linear approximation of the objective and a convex regularizer that keeps the step small)
- Gradient descent suffices for the optimal regret rate; the right regularizer gives the right dependence on the dimension.
Continuous Allocations

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

(Recall: $T$ is number of rounds of the game; $K$ is number of venues.)
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 $V \sqrt{T \ln K}/16$.

(Recall: $T$ is number of rounds of the game; $K$ is number of venues.)
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 $\tilde{O}((VTK)^{2/3})$.

**Theorem:**
Any algorithm has regret $\tilde{\Omega}((VTK)^{1/2})$.

(Value of the game is $O(T^{1/2})$; no known algorithm.)
Dark Pools

- Allow adversarial choice of volumes and transactions.
Dark Pools

- Allow adversarial choice of volumes and transactions.
- Per trial regret rate superior to previous best known bounds for probabilistic setting.
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.
Outline

- Decision problems as sequential games
  1. Allocation to dark pools
  2. Pricing options
  3. Linear regression
Given a financial contract with a known payoff at a future time $T$, how much is it worth now?
Option Pricing

Joint work with Jacob Abernethy, Rafael Frongillo, Andre Wibisono

- Given a financial contract with a known payoff at a future time \( T \), how much is it worth now?
- **European call / put option**: contract that gives the holder the **right** to buy / sell an asset at **strike price** \( K \) at **expiration time** \( T \)

**Payoff of call option:**
\[
g_C(S_T) = \max\{0, S_T - K\}
\]

**Payoff of put option:**
\[
g_P(S_T) = \max\{0, K - S_T\}
\]
Assume **no arbitrage**: No opportunity to make riskless profit
Option Pricing

- Assume **no arbitrage**: No opportunity to make riskless profit
- **Black-Scholes (1973)**: Asset price $S_t \sim$ geometric Brownian motion

\[
\log S_t = \log S_0 + \sigma B_t + \left( \mu - \frac{\sigma^2}{2} \right) t
\]

Multiplicative price fluctuation is normally distributed

\[
S_{t+\Delta t} - S_t = r S_t
\]

\[
r \approx \log (1 + r) \sim \mathcal{N} \left( \left( \mu - \frac{\sigma^2}{2} \right) \Delta t, \sigma^2 \Delta t \right)
\]
Option Pricing

- **Assume no arbitrage:** No opportunity to make riskless profit
- **Black-Scholes (1973):** Asset price $S_t \sim$ geometric Brownian motion

$$\log S_t = \log S_0 + \sigma B_t + \left(\mu - \frac{\sigma^2}{2}\right) t$$

Multiplicative price fluctuation is normally distributed

$$S_{t+\Delta t} - S_t = r S_t$$

$$r \approx \log(1 + r) \sim N\left(\left(\mu - \frac{\sigma^2}{2}\right) \Delta t, \sigma^2 \Delta t\right)$$

- **Hedging strategy:** Trade underlying asset to replicate option payoff
Option Pricing

- Option value is \( V(S, t) \) when asset price is \( S \) at time \( t \)
Option Pricing

- Option value is $V(S, t)$ when asset price is $S$ at time $t$
- Black-Scholes strategy: invest $\Delta(S, t) = S \frac{\partial V}{\partial S}(S, t)$ in asset at time $t$
Option Pricing

- Option value is $V(S, t)$ when asset price is $S$ at time $t$
- Black-Scholes strategy: invest $\Delta(S, t) = S \frac{\partial V}{\partial S}(S, t)$ in asset at time $t$
- Option value $V(S, t)$ satisfies (logarithmic) heat equation

$$V_t(S, t) + \frac{1}{2} S^2 V_{SS}(S, t) = 0$$

with boundary condition given by option payoff $V(S, T) = g(S)$
Option Pricing

- Option value is \( V(S, t) \) when asset price is \( S \) at time \( t \)
- Black-Scholes strategy: invest \( \Delta(S, t) = S V_S(S, t) \) in asset at time \( t \)
- Option value \( V(S, t) \) satisfies (logarithmic) \textit{heat equation}

\[
V_t(S, t) + \frac{1}{2} S^2 V_{SS}(S, t) = 0
\]

with boundary condition given by option payoff \( V(S, T) = g(S) \)

\[\text{Black-Scholes Formula:} \]

\[ V(S, t) = \mathbb{E}[g(S \cdot G(T - t))] \]

where \( G(t) \sim \text{GBM}(0, \sigma^2) \)
Adversarial Option Pricing

- Black-Scholes requires strong assumption on $S_t$
Adversarial Option Pricing

- Black-Scholes requires strong assumption on $S_t$
- Can we construct trading strategy robust to adversarially chosen price?
Adversarial Option Pricing

- Black-Scholes requires strong assumption on $S_t$
- Can we construct trading strategy robust to adversarially chosen price?
- DeMarzo, Kremer, Mansour (2006):
  - Trading algorithm with lower bound on payoff $\Rightarrow$ upper bound on option price
Adversarial Option Pricing

- Our approach: option pricing from **online learning** perspective
Our approach: option pricing from **online learning** perspective

Sequential zero-sum online trading game between Investor and Market
Adversarial Option Pricing

- Our approach: option pricing from **online learning** perspective
- Sequential zero-sum online trading game between Investor and Market
- Suppose there are $n$ trading periods before expiration time $T$
Adversarial Option Pricing

- Our approach: option pricing from **online learning** perspective
- Sequential zero-sum online trading game between Investor and Market
- Suppose there are $n$ trading periods before expiration time $T$

**Investor**
- Observes asset price $S$
- Invests $\Delta$

**Market**
- Selects fluctuation $r$
- Updates price $S \leftarrow S(1 + r)$

Investor profits $\Delta r$
Minimax regret is “minimax option price”
Minimax regret is “**minimax option price**”

How much more money Investor could have made from option:

\[
\text{regret} = g\left(S \cdot \prod_{i=1}^{n} (1 + r_i)\right) - \sum_{i=1}^{n} \Delta_i r_i
\]

\[
\text{option payoff} - \text{trading profit}
\]

\[
V_\zeta^n(S, c) = \inf_{\Delta_1} \sup_{r_1} \cdots \inf_{\Delta_n} \sup_{r_n} g\left(S \cdot \prod_{i=1}^{n} (1 + r_i)\right) - \sum_{i=1}^{n} \Delta_i r_i
\]

with **cumulative volatility constraint:**

\[
\sum_{i=1}^{n} r_i^2 \leq c
\]

**maximum jump constraint:**

\[
|r_i| \leq \zeta_n
\]
Convergence to Black-Scholes Price

**Theorem (Lower Bound):**
If payoff function $g$ is Lipschitz and $\lim\inf_{n \to \infty} n \zeta_n^2 > c$, then
$$\lim\inf_{n \to \infty} V_n^\zeta(S, c) \geq U(S, c)$$
Define **Black-Scholes price**: \( U(S, c) = \mathbb{E}[g(S \cdot G(c))] \)

**Theorem (Lower Bound):**
If payoff function \( g \) is Lipschitz and \( \lim \inf_{n \to \infty} n \zeta_n^2 > c \), then
\[
\lim \inf_{n \to \infty} V_{\zeta_n}^n(S, c) \geq U(S, c)
\]
Convergence to Black-Scholes Price

- Let $G(t) \overset{d}{=} \exp(B(t) - \frac{1}{2}t)$ be GBM with zero drift and unit volatility.
- Define **Black-Scholes price**: $U(S, c) = \mathbb{E}[g(S \cdot G(c))]$

**Theorem (Lower Bound):**
If payoff function $g$ is Lipschitz and $\lim inf_{n \to \infty} n \zeta_n^2 > c$, then
$$\lim inf_{n \to \infty} V_{\zeta_n}^n(S, c) \geq U(S, c)$$
Convergence to Black-Scholes Price

- Let \( G(t) \overset{d}{=} \exp(B(t) - \frac{1}{2} t) \) be GBM with zero drift and unit volatility.
- Define **Black-Scholes price**: \( U(S, c) = \mathbb{E}[g(S \cdot G(c))] \)

**Theorem (Lower Bound):**
If payoff function \( g \) is Lipschitz and \( \lim_{n \to \infty} n \zeta_n^2 > c \), then
\[
\lim_{n \to \infty} V^n_{\zeta_n}(S, c) \geq U(S, c)
\]

**Theorem (Upper Bound):**
If \( g \) is convex, \( L \)-Lipschitz, and \( K \)-eventually linear, then for any \( \zeta > 0 \),
\[
V^n_{\zeta}(S, c) \leq U(S, c) + 18LK c \zeta^{1/4}
\]

**Corollary:**
If also \( \zeta_n \to 0 \), then
\[
\lim_{n \to \infty} V^n_{\zeta_n}(S, c) = U(S, c)
\]
Let $G(t) \overset{d}{=} \exp(B(t) - \frac{1}{2} t)$ be GBM with zero drift and unit volatility.

Define **Black-Scholes price**: $U(S, c) = \mathbb{E}[g(S \cdot G(c))]$

**Theorem (Lower Bound):**
If payoff function $g$ is Lipschitz and $\lim \inf_{n \to \infty} n \zeta_n^2 > c$, then
$$\lim \inf_{n \to \infty} V_{\zeta_n}^n (S, c) \geq U(S, c)$$

**Theorem (Upper Bound):**
If $g$ is convex, $L$-Lipschitz, and $K$-eventually linear, then for any $\zeta > 0$,
$$V_{\zeta}^n (S, c) \leq U(S, c) + 18LK c \zeta^{1/4}$$

**Corollary:**
If also $\zeta_n \to 0$, then $\lim_{n \to \infty} V_{\zeta_n}^n (S, c) = U(S, c)$
Let $G(t) \stackrel{d}{=} \exp(B(t) - \frac{1}{2} t)$ be GBM with zero drift and unit volatility.

Define **Black-Scholes price**: $U(S, c) = \mathbb{E}[g(S \cdot G(c))]$

**Theorem (Lower Bound):**
If payoff function $g$ is Lipschitz and $\lim \inf_{n \to \infty} n \zeta_n^2 > c$, then
$$
\lim \inf_{n \to \infty} V_n^n(S, c) \geq U(S, c)
$$

**Theorem (Upper Bound):**
If $g$ is convex, $L$-Lipschitz, and $K$-eventually linear, then for any $\zeta > 0$,
$$
V_\zeta^n(S, c) \leq U(S, c) + 18LK c \zeta^{1/4}
$$

**Corollary:**
If also $\zeta_n \to 0$, then $\lim_{n \to \infty} V_\zeta^n(S, c) = U(S, c)$

Black-Scholes as “worst-case” model
The upper bound is obtained by considering the Black-Scholes strategy for Investor:

\[ \Delta(S, c) = S U_S(S, c) \]
The upper bound is obtained by considering the **Black-Scholes strategy** for Investor:

\[ \Delta(S, c) = S U_S(S, c) \]

**Lower bound proof sketch:**
- Analyze randomized price for Market: \( R_{i,n} \sim \text{Uniform}\{\pm \sqrt{c/n}\} \) i.i.d.
- Central limit theorem:
  \[ \mathbb{E}[g(S \prod_{i=1}^{n}(1 + R_{i,n}))] \to \mathbb{E}[g(S \cdot G(c))] = U(S, c) \]
• Decision problems as sequential games
  1 Allocation to dark pools
  2 Pricing options
  3 Linear regression
Online fixed design linear regression

Joint work with Wouter Koolen, Alan Malek, Eiji Takimoto, Manfred Warmuth.
Online fixed design linear regression

Joint work with Wouter Koolen, Alan Malek, Eiji Takimoto, Manfred Warmuth.

**Protocol**

Given:

- $T$
- $x_1, \ldots, x_T \in \mathbb{R}^p$
- $Y_T \subset \mathbb{R}^T$

For $t = 1, 2, \ldots, T$:

1. Learner predicts $\hat{y}_t \in \mathbb{R}$
2. Adversary reveals $y_t \in \mathbb{R}$
3. Learner incurs loss $(\hat{y}_t - y_t)^2$

Regret:

$$T \sum_{t=1}^T (\hat{y}_t - y_t)^2 - \min_{\beta \in \mathbb{R}^p} T \sum_{t=1}^T (\beta^T x_t - y_t)^2$$
Joint work with Wouter Koolen, Alan Malek, Eiji Takimoto, Manfred Warmuth.

Protocol

Given: $T$;

\[\text{Regret} = \sum_{t=1}^{T} (\hat{y}_t - y_t)^2 - \min_{\beta \in \mathbb{R}^p} \sum_{t=1}^{T} (\beta^\top x_t - y_t)^2.\]
Joint work with Wouter Koolen, Alan Malek, Eiji Takimoto, Manfred Warmuth.

Protocol

Given: $T; x_1, \ldots, x_T \in \mathbb{R}^p$;
Online fixed design linear regression

Joint work with Wouter Koolen, Alan Malek, Eiji Takimoto, Manfred Warmuth.

Protocol

Given: $T; x_1, \ldots, x_T \in \mathbb{R}^p; \mathcal{Y}^T \subset \mathbb{R}^T.$
Online fixed design linear regression

Joint work with Wouter Koolen, Alan Malek, Eiji Takimoto, Manfred Warmuth.

Protocol

Given: \( T; x_1, \ldots, x_T \in \mathbb{R}^p; \mathcal{Y}^T \subset \mathbb{R}^T \).

For \( t = 1, 2, \ldots, T \):

- Learner predicts \( \hat{y}_t \in \mathbb{R} \).
- Adversary reveals \( y_t \in \mathcal{Y}_T \).
- Learner incurs loss \((\hat{y}_t - y_t)^2\).

Regret = \[ T \sum_{t=1}^{T} (\hat{y}_t - y_t)^2 - \min_{\beta \in \mathbb{R}^p} T \sum_{t=1}^{T} (\beta^\top x_t - y_t)^2 \].
Online fixed design linear regression

Joint work with Wouter Koolen, Alan Malek, Eiji Takimoto, Manfred Warmuth.

### Protocol

Given: \( T; x_1, \ldots, x_T \in \mathbb{R}^p; Y^T \subset \mathbb{R}^T \).

For \( t = 1, 2, \ldots, T \):

- Learner predicts \( \hat{y}_t \in \mathbb{R} \)
Online fixed design linear regression

Joint work with Wouter Koolen, Alan Malek, Eiji Takimoto, Manfred Warmuth.

**Protocol**

Given: \( T; \mathbf{x}_1, \ldots, \mathbf{x}_T \in \mathbb{R}^p; \mathcal{Y}^T \subseteq \mathbb{R}^T \).

For \( t = 1, 2, \ldots, T \):

- Learner predicts \( \hat{y}_t \in \mathbb{R} \)
- Adversary reveals \( y_t \in \mathbb{R} \)
Online fixed design linear regression

Joint work with Wouter Koolen, Alan Malek, Eiji Takimoto, Manfred Warmuth.

Protocol

Given: \( T; x_1, \ldots, x_T \in \mathbb{R}^p; Y^T \subset \mathbb{R}^T \).
For \( t = 1, 2, \ldots, T \):

- Learner predicts \( \hat{y}_t \in \mathbb{R} \)
- Adversary reveals \( y_t \in \mathbb{R} \) (\( y_1^T \in Y^T \))

\[
\text{Regret} = \sum_{t=1}^{T} (\hat{y}_t - y_t)^2 - \min_{\beta \in \mathbb{R}^p} \sum_{t=1}^{T} (\beta^\top x_t - y_t)^2.
\]
Online fixed design linear regression

Joint work with Wouter Koolen, Alan Malek, Eiji Takimoto, Manfred Warmuth.

Protocol

Given: $T; x_1, \ldots, x_T \in \mathbb{R}^p; \mathcal{Y}^T \subset \mathbb{R}^T$. For $t = 1, 2, \ldots, T$:

- Learner predicts $\hat{y}_t \in \mathbb{R}$
- Adversary reveals $y_t \in \mathbb{R}$ ($y_1^T \in \mathcal{Y}^T$)
- Learner incurs loss $(\hat{y}_t - y_t)^2$. 

Regret = $T \sum_{t=1}^{T} (\hat{y}_t - y_t)^2 - \min_{\beta \in \mathbb{R}^p} T \sum_{t=1}^{T} (\beta \top x_t - y_t)^2$. 
Online fixed design linear regression

Joint work with Wouter Koolen, Alan Malek, Eiji Takimoto, Manfred Warmuth.

### Protocol

**Given:** \( T; x_1, \ldots, x_T \in \mathbb{R}^p; \mathcal{Y}^T \subset \mathbb{R}^T. \)

For \( t = 1, 2, \ldots, T \):

- Learner predicts \( \hat{y}_t \in \mathbb{R} \)
- Adversary reveals \( y_t \in \mathbb{R} \) \((y_1^T \in \mathcal{Y}^T)\)
- Learner incurs loss \((\hat{y}_t - y_t)^2\).

\[
\text{Regret} = \sum_{t=1}^{T} (\hat{y}_t - y_t)^2 - \min_{\beta \in \mathbb{R}^p} \sum_{t=1}^{T} \left( \beta^T x_t - y_t \right)^2.
\]
Online fixed design linear regression

**Online linear regression: previous work**

- (Foster, 1991): $\ell_2$-regularized least squares.
- (Cesa-Bianchi et al, 1996): $\ell_2$-constrained least squares.
- (Kivinen and Warmuth, 1997): exponentiated gradient (relative entropy regularization).
- (Forster, 1999; Azoury and Warmuth, 2001): aggregating algorithm is last-step minimax.
Online fixed design linear regression

Online linear regression: previous work

- (Foster, 1991): $\ell_2$-regularized least squares.
- (Cesa-Bianchi et al, 1996): $\ell_2$-constrained least squares.
- (Kivinen and Warmuth, 1997): exponentiated gradient (relative entropy regularization).
- (Forster, 1999; Azoury and Warmuth, 2001): aggregating algorithm is last-step minimax.

This work

- The optimal strategy.
Linear regression in a probabilistic setting

Ordinary least squares (linear model, uncorrelated errors)

Given \((x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^p \times \mathbb{R}\),

\[
\hat{\beta} = \left( \sum_{t=1}^n x_t x_t^\top \right)^{-1} \sum_{t=1}^n x_t y_t,
\]

and for a subsequent \(x \in \mathbb{R}^p\), predict

\[
\hat{y} = x^\top \hat{\beta} = x^\top \left( \sum_{t=1}^n x_t x_t^\top \right)^{-1} \sum_{t=1}^n x_t y_t.
\]
### Ordinary least squares (linear model, uncorrelated errors)

Given \((x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^p \times \mathbb{R}\), choose

$$
\hat{\beta} = \left( \sum_{t=1}^{n} x_t x_t^\top \right)^{-1} \sum_{t=1}^{n} x_t y_t,
$$

A sequential version of OLS

$$
\hat{y}_{n+1} = x_{n+1}^\top \left( \sum_{t=1}^{n} x_t x_t^\top \right)^{-1} \sum_{t=1}^{n} x_t y_t.
$$
Ordinary least squares (linear model, uncorrelated errors)

Given \((x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^p \times \mathbb{R}\), choose

\[
\hat{\beta} = \left(\sum_{t=1}^{n} x_t x_t^\top\right)^{-1} \sum_{t=1}^{n} x_t y_t,
\]

and for a subsequent \(x \in \mathbb{R}^p\), predict

\[
\hat{y} = x^\top \hat{\beta} = x^\top \left(\sum_{t=1}^{n} x_t x_t^\top\right)^{-1} \sum_{t=1}^{n} x_t y_t,
\]
Linear regression in a probabilistic setting

Ordinary least squares (linear model, uncorrelated errors)

Given \((x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^p \times \mathbb{R}\), choose

\[
\hat{\beta} = \left( \sum_{t=1}^{n} x_t x_t^T \right)^{-1} \sum_{t=1}^{n} x_t y_t,
\]

and for a subsequent \(x \in \mathbb{R}^p\), predict

\[
\hat{y} = x^\top \hat{\beta} = x^\top \left( \sum_{t=1}^{n} x_t x_t^T \right)^{-1} \sum_{t=1}^{n} x_t y_t,
\]

A sequential version of OLS

\[
\hat{y}_{n+1} := x_{n+1}^\top \left( \sum_{t=1}^{n} x_t x_t^T \right)^{-1} \sum_{t=1}^{n} x_t y_t.
\]
Linear regression in a probabilistic setting

Ordinary least squares (linear model, uncorrelated errors)

Given \((x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^p \times \mathbb{R}\), choose

\[
\hat{\beta} = \left( \sum_{t=1}^{n} x_t x_t^\top \right)^{-1} \sum_{t=1}^{n} x_t y_t,
\]

and for a subsequent \(x \in \mathbb{R}^p\), predict

\[
\hat{y} = x^\top \hat{\beta} = x^\top \left( \sum_{t=1}^{n} x_t x_t^\top \right)^{-1} \sum_{t=1}^{n} x_t y_t.
\]

A sequential version of ridge regression

\[
\hat{y}_{n+1} := x_{n+1}^\top \left( \sum_{t=1}^{n} x_t x_t^\top + \lambda I \right)^{-1} \sum_{t=1}^{n} x_t y_t.
\]
Fix $x_1, \ldots, x_T \in \mathbb{R}^p$. 

Use sufficient statistics:

$$s_n = \sum_{t=1}^n y_t x_t.$$ 

Provided:

$$B_n \geq n - 1 \sum_{t=1}^n |x_t^\top x_t|.$$ 

Minimax strategy: linear

$$\hat{y}_{n+1} = x_{n+1}^\top s_n.$$ 

$$P_{n+1} = n \sum_{t=1}^n x_t x_t^\top + 1 + x_{n+1}^\top P_{n+1} x_{n+1} x_{n+1}^\top.$$
Fix \( x_1, \ldots, x_T \in \mathbb{R}^p \).

\[ y^T = \{(y_1, \ldots, y_T) : |y_t| \leq B_t\} \]
Online fixed design linear regression

**Sufficient statistics**

Fix $x_1, \ldots, x_T \in \mathbb{R}^p$.

Use sufficient statistics: $s_n = \sum_{t=1}^{n} y_t x_t$.

$Y^T = \{(y_1, \ldots, y_T) : |y_t| \leq B_t\}$.
Online fixed design linear regression

**Sufficient statistics**

Fix $x_1, \ldots, x_T \in \mathbb{R}^p$.

Use sufficient statistics: $s_n = \sum_{t=1}^{n} y_t x_t$.

$\mathcal{Y}^T = \{(y_1, \ldots, y_T) : |y_t| \leq B_t\}$.

**Minimax* strategy: linear**

$\hat{y}_{n+1}^* = x_{n+1}^\top P_{n+1} s_n$. 

$P_{n+1}^{-1} = n^{-1} \sum_{t=1}^{n} x_t x_t^\top + T^{-1} \sum_{t=n+1}^{T} x_t x_t^\top$.
Online fixed design linear regression

Sufficient statistics

Fix $x_1, \ldots, x_T \in \mathbb{R}^p$. Use sufficient statistics: $s_n = \sum_{t=1}^{n} y_t x_t$.

$\mathcal{Y}^T = \{(y_1, \ldots, y_T) : |y_t| \leq B_t\}$.

Minimax* strategy: linear

$$\hat{y}_{n+1}^* = x_{n+1}^\top P_{n+1} s_n.$$
Online fixed design linear regression

**Sufficient statistics**

Fix \( x_1, \ldots, x_T \in \mathbb{R}^p \).

Use sufficient statistics: \( s_n = \sum_{t=1}^{n} y_t x_t \).

\( \mathcal{Y}^T = \{(y_1, \ldots, y_T) : |y_t| \leq B_t \} \).

**Minimax* strategy: linear**

\[
\hat{y}_{n+1}^* = x_{n+1}^T P_{n+1} s_n.
\]

\[
P_{n}^{-1} = \sum_{t=1}^{n} x_t x_t^\top + \sum_{t=n+1}^{T} \frac{x_t^\top P_t x_t}{1 + x_t^\top P_t x_t} x_t x_t^\top.
\]
Online fixed design linear regression

**Sufficient statistics**

Fix $x_1, \ldots, x_T \in \mathbb{R}^p$.

Use sufficient statistics: $s_n = \sum_{t=1}^{n} y_t x_t$.

$\mathcal{Y}^T = \{(y_1, \ldots, y_T) : |y_t| \leq B_t\}$.

* provided: $B_n \geq \sum_{t=1}^{n-1} |x_n^\top P_n x_t| B_t$.

**Minimax* strategy: linear**

$\hat{y}_{n+1}^* = x_{n+1} P_{n+1} s_n$.

$P_{n}^{-1} = \sum_{t=1}^{n} x_t x_t^\top + \sum_{t=n+1}^{T} \frac{x_t^\top P_t x_t}{1 + x_t^\top P_t x_t} x_t x_t^\top$. 
Linear regression

Box constraints

\[ \gamma^T = \{(y_1, \ldots, y_T) : |y_n| \leq B_n \} \]

\[ B_n \geq \sum_{t=1}^{n-1} |x_n^\top P_n x_t| B_t. \]
Linear regression

Box constraints

\[ \mathcal{Y}_T = \{(y_1, \ldots, y_T) : |y_n| \leq B_n\} \]

\[ B_n \geq \sum_{t=1}^{n-1} \left| x_n^\top P_n x_t \right| B_t. \]

Minimax strategy: linear

\[ \hat{y}^*_n = x_n^\top P_n s_{n-1}. \]
Linear regression

Box constraints

\[ \mathcal{Y}^T = \{(y_1, \ldots, y_T) : |y_n| \leq B_n\} \]

\[ B_n \geq \sum_{t=1}^{n-1} \left| x_n^\top P_n x_t \right| B_t. \]

Minimax strategy: linear

\[ \hat{y}_n^* = x_n^\top P_n s_{n-1}. \]

Optimal shrinkage

\[ P_n^{-1} = \sum_{t=1}^{n} x_t x_t^\top + \sum_{t=n+1}^{T} \frac{x_t^\top P_t x_t}{1 + x_t^\top P_t x_t} x_t x_t^\top. \]
Linear regression

**Box constraints**

\[ \mathcal{Y}^T = \{(y_1, \ldots, y_T) : |y_n| \leq B_n\} \]

\[ B_n \geq \sum_{t=1}^{n-1} \left| x_n^\top P_n x_t \right| B_t. \]

**Minimax strategy: linear**

\[ \hat{y}_n^* = x_n^\top P_n s_{n-1}. \]

**Regret**

\[ \text{Regret} = \sum_{t=1}^T B_t^2 x_t^\top P_t x_t. \]

\[ P_n^{-1} = \sum_{t=1}^n x_t x_t^\top + \sum_{t=n+1}^T \frac{x_t^\top P_t x_t}{1 + x_t^\top P_t x_t} x_t x_t^\top. \]

c.f. ridge regression:

\[ \sum_{t=1}^n x_t x_t^\top + \lambda I. \]
Linear regression

Box constraints

\[ \mathcal{Y}^T = \{ (y_1, \ldots, y_T) : |y_n| \leq B_n \} \]

\[ B_n \geq \sum_{t=1}^{n-1} \left| x_n^\top P_n x_t \right| B_t. \]

Regret

\[ \text{Regret} = \sum_{t=1}^{T} B_t^2 x_t^\top P_t x_t. \]

Minimax strategy: linear

\[ \hat{y}_n^* = x_n^\top P_n s_{n-1}. \]

Optimal shrinkage

\[ P_n^{-1} = \sum_{t=1}^{n} x_t x_t^\top + \sum_{t=n+1}^{T} \frac{x_t^\top P_t x_t}{1 + x_t^\top P_t x_t} x_t x_t^\top. \]

C.f. ridge regression:

\[ \sum_{t=1}^{n} x_t x_t^\top + \lambda I. \]
### Linear regression

#### Box constraints

\[
T = \{(y_1, \ldots, y_T) : |y_n| \leq B_n\}
B_n \geq \sum_{t=1}^{n-1} \left| x_n^\top P_n x_t \right| B_t.
\]

#### Regret

\[
\text{Regret} = \sum_{t=1}^{T} B_t^2 x_t^\top P_t x_t.
\]

#### Minimax strategy: linear

\[
\hat{y}_n^* = x_n^\top P_n s_{n-1}.
\]

#### Optimal shrinkage

\[
P_{n}^{-1} = \sum_{t=1}^{n} x_t x_t^\top + \sum_{t=n+1}^{T} \frac{x_t^\top P_t x_t}{1 + x_t^\top P_t x_t} x_t x_t^\top.
\]

c.f. ridge regression:

\[
\sum_{t=1}^{n} x_t x_t^\top + \lambda I.
\]
Linear regression

Box constraints

\[ \mathcal{Y}^T = \{(y_1, \ldots, y_T) : |y_n| \leq B_n\} \]

\[ B_n \geq \sum_{t=1}^{n-1} |x_n^\top P_n x_t| B_t. \]

Minimax strategy: linear

\[ \hat{y}_n^* = x_n^\top P_n s_{n-1}. \]

Optimal shrinkage

\[ P_n^{-1} = \sum_{t=1}^{n} x_t x_t^\top + \sum_{t=n+1}^{T} \frac{x_t^\top P_t x_t}{1 + x_t^\top P_t x_t} x_t x_t^\top. \]

c.f. ridge regression:

\[ \sum_{t=1}^{n} x_t x_t^\top + \lambda I. \]
Linear regression with adversarial covariates

Legal covariate sequences

For any $t \geq 0$, any $x_1, \ldots, x_t$ and any $P_t$, the following two conditions are equivalent.

1. There is a $T \geq t$ and a sequence $x_{t+1}, \ldots, x_T$ such that

   $$P_T^{-1} = \sum_{q=1}^T x_q x_q^\top.$$

2. $P_t^{-1} \succeq \sum_{q=1}^t x_q x_q^\top$.

Adversarial covariates

Thus, each $P_0 \succeq 0$ (a ‘covariance budget’) defines a set of sequences $x_1, \ldots, x_T$ (and corresponding suitable bounds on $y_1, \ldots, y_T$). The same strategy is optimal for each of these sequences.
Linear regression

\[ \hat{y}_n^* = x_n^\top P_n s_{n-1} \]

- Minimax optimal for two families of label constraints: box constraints and problem-weighted \( \ell_2 \) norm constraints.
Linear regression

\[ \hat{y}_n^* = x_n^\top P_n s_{n-1} \]

- Minimax optimal for two families of label constraints: box constraints and problem-weighted $\ell_2$ norm constraints.
- Strategy does not need to know the constraints.
\[ \hat{y}_n^* = x_n^\top P_n s_{n-1} \]

- Minimax optimal for two families of label constraints: box constraints and problem-weighted $\ell_2$ norm constraints.
- Strategy does not need to know the constraints.
- Regret is $O(p \log T)$. 
Linear regression

\[ \hat{y}_n^* = x_n^\top P_n s_{n-1} \]

- Minimax optimal for two families of label constraints: box constraints and problem-weighted $\ell_2$ norm constraints.
- Strategy does not need to know the constraints.
- Regret is $O(p \log T)$.
- Same strategy is optimal for covariate sequences consistent with some ‘covariance budget’ $P_0$. 
Other games with efficient minimax optimal strategies

Euclidean loss

- Prediction in $\mathbb{R}^d$: $\mathcal{Y} \subseteq \mathbb{R}^d$, $\mathcal{A} = \mathbb{R}^d$, Euclidean loss: $\ell(\hat{y}, y) = \frac{1}{2} \| \hat{y} - y \|^2$.

- Minimax strategy is empirical minimizer plus shrinkage towards center of smallest ball containing $\mathcal{Y}$: $a_{t+1}^* = t\alpha_{t+1}\bar{y}_t + (1 - t\alpha_{t+1})c$.

- Regret:

$$\frac{r^2}{2} \sum_{t=1}^{T} \alpha_t,$$

where $r$ is radius of smallest ball,

$$\alpha_T = \frac{1}{T}, \quad \alpha_t = \alpha_{t+1}^2 + \alpha_{t+1}$$
Other games with efficient minimax optimal strategies

Time series forecasting

\[
\min_{a_1} \max_{x_1} \cdots \min_{a_T} \max_{x_T} \sum_{t=1}^{T} ||a_t - x_t||^2
\]

\[
\min_{\hat{a}_1, \ldots, \hat{a}_T} \sum_{t=1}^{T} ||\hat{a}_t - x_t||^2 + \lambda \sum_{t=1}^{T+1} ||\hat{a}_t - \hat{a}_{t-1}||^2.
\]

- Expression for regret when \( x_t \) bounded. (And a bound when it is not.)
- Minimax strategy makes linear predictions.
- Regret is \( O\left(\frac{T}{\sqrt{1 + \lambda}}\right) \).
- More generally, penalize comparator by the energy of the innovations of a time series model. Efficient linear minimax strategy. Regret?
Decision problems as sequential games
1 Allocation to dark pools
2 Pricing options
3 Linear regression
Outline

- Decision problems as sequential games
  1 Allocation to dark pools
  2 Pricing options
  3 Linear regression

Formulating decision problems as sequential games

- Decision problems: regression, classification, order allocation, dynamic pricing, portfolio optimization, option pricing.
- Rather than model the process generating the data probabilistically, we view it as an adversary.
Formulating decision problems as sequential games

- Decision problems: regression, classification, order allocation, dynamic pricing, portfolio optimization, option pricing.
- Rather than model the process generating the data probabilistically, we view it as an adversary.

Decision-making = hedging against the future choices of the process generating the data.