Tutorial: PART 1

Online Convex Optimization, A Game-Theoretic Approach to Learning


Elad Hazan
Princeton University

Satyen Kale
Yahoo Research
Agenda

1. Motivating examples
2. Unified framework & why it makes sense
3. Algorithms & main results / techniques
4. Research directions & open questions
Section 1: Motivating Examples

Inherently adversarial & online
Sequential Spam Classification - online & adversarial learning

- observe n features (words): $a_t \in R^n$
- Predict label $\hat{b}_t \in \{-1,1\}$, feedback $b_t$
- Objective: average error $\rightarrow$ best linear model in hindsight

$$\min_{x \in R^d} \sum_t \log(1 + e^{-b_t \cdot x^T a_t})$$
Spam Classifier Selection

Stream of emails:

\[ a \in \mathbb{R}^d \]

Objective: make predictions with accuracy \( \rightarrow \) best classifier in hindsight

\[ \log(1 + e^{-b \cdot x^T a}) + \lambda_1 |x|^2 \]

\[ \log(1 + e^{-b \cdot x^T a}) + \lambda_2 |x|^2 \]

\[ \log(1 + e^{-b \cdot x^T a}) + \lambda_n |x|^2 \]
Universal portfolio selection

Price relatives:
\[ r_t(i) = \frac{\text{closing price}}{\text{opening price}} \]
for commodity \( i \)

Distribution of wealth \( x_t \)

\[ \begin{align*}
   & r_t \times 1.5 \quad x_t \times 1.5 \\
   & \times 1 \quad 0 \end{align*} \]

Wealth multiplier =
\[ (\frac{1}{2} \times 1.5 + 0 \times 1 + \frac{1}{2} \times 1 + 0 \times \frac{1}{2}) = 1 \frac{1}{4} \]

Market - No Statistical assumptions

\[ \log W_{t+1} = \log W_t + \log(r_t^T x_t) \]

Objective: choose portfolios s.t. avg. log wealth \( \rightarrow \) avg. log wealth of best CRP
Constant rebalancing portfolio

- Single currency depreciates exponentially
- The 50/50 Constant Rebalanced Portfolio (CRP) makes 5% every day
### Recommendation Systems

**Objective:** ratings with accuracy → best low rank matrix

**Computation challenge:** optimizing over low rank matrices

---

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>1</td>
<td>1</td>
<td>X</td>
<td>...</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>...</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>-1</td>
<td>X</td>
<td>...</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>1</td>
<td>X</td>
<td>...</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>-1</td>
<td>X</td>
<td>X</td>
<td>...</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>...</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>1</td>
<td>X</td>
<td>X</td>
<td>...</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>-1</td>
<td>...</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
Ad Selection

• Yahoo, Google etc display ads on pages, clicks generate revenue

• Abstraction:
  • Observe a stream of users
  • Given a user, select one ad to display
  • Feedback: click (or not) on the ad shown

• Objective: average revenue per ad \(\rightarrow\) best policy in a given class

• Additional difficulty: partial (bandit) feedback
  (feedback only for ad actually shown)
2. Methodology: Online Convex Optimization
   definition & relation to PAC learning
go through examples
argue it makes sense
Statistical (PAC) learning

Nature: i.i.d from distribution $D$ over $A \times B = \{(a, b)\}$

Learner:

Hypothesis $h$

Loss, e.g. $\ell(h, (a, b)) = (h(a) - b)^2$

$$\text{err}(h) = \mathbb{E}_{a, b \sim D}[\ell(h, (a, b))]$$

Hypothesis class $H: X \to Y$ is learnable if $\forall \epsilon, \delta > 0$ exists algorithm s.t. after seeing $m$ examples, for $m = poly(\delta, \epsilon, \text{dimension}(H))$ finds $h$ s.t. w.p. $1 - \delta$:

$$\text{err}(h) \leq \min_{h^* \in H} \text{err}(h^*) + \epsilon$$
Online Learning in Games

Iteratively, for $t = 1, 2, ..., T$

Player: $h_t \in H$

Adversary: $(a_t, b_t) \in A$

Loss $\ell(h_t, (a_t, b_t))$

Goal: minimize (average, expected) regret:

$$\frac{1}{T} \left[ \sum_t \ell(h_t, (a_t, b_t)) - \min_{h^* \in H} \sum_t \ell(h^*, (a_t, b_t)) \right] \quad \xrightarrow{T \to \infty} \quad 0$$

Can we learn in games efficiently?  Regret $\rightarrow o(T)$?

(i.e. as fundamental theorem of stat. learning)
Online vs. Batch (PAC, Statistical)

**Online convex optimization & regret minimization**
- Learning against an adversary
- Regret
- Streaming setting: process examples sequentially
- Less assumptions, stronger guarantees
- Harder algorithmically

**Batch learning, PAC/statistical learning**
- Nature is benign, i.i.d. samples from unknown distribution
- Generalization error
- Batch: process dataset multiple times
- Weaker guarantees (changing distribution, etc.)
- Easier algorithmically
Online Convex Optimization

Adversary

Convex cost function $f_t$

Point $x_t$ in convex set $K$ in $\mathbb{R}^n$

Online Player

Access to $K, f_t$?

Total loss $\sum_t f_t(x_t)$

$$\text{Regret} = \sum_t f_t(x_t) - \min_{x^* \in K} \sum_t f_t(x^*)$$
Prediction from expert advice

• Decision set = set of all distributions over n experts

\[ K = \Delta_n = \{ x \in \mathbb{R}^n , \sum_i x_i = 1 , x_i \geq 0 \} \]

• Cost functions?
  Let: \( c_t(i) = \text{loss of expert } i \text{ in round } t \)

\[ f_t(x) = c_t^\top x = \sum_i c_t(i)x(i) \]

\( f_t(x) \) = expected loss if choosing experts by distribution \( x_t \)

\( \text{Regret} = \text{difference in } \# \text{ expected loss vs. best expert} : \)

\[ \text{Regret} = \sum_t f_t(x_t) - \min_{x^* \in K} \sum_t f_t(x^*) \]
Parameterized Hypothesis Classes

• $H$ - parameterized by vector $x$ in convex set $K \subseteq R^n$
• In round $t$, define loss function based on example $(a_t, b_t)$

1. **Online Linear Spam Filtering:**
   • $K = \{x \mid \| x \| \leq \omega\}$
   • Loss function $f_t(x) = (a_t^T x - b_t)^2$

2. **Online Soft Margin SVM:**
   • $K = R^n$
   • Loss function $f_t(x) = \max\{0, 1 - b_t a_t^T x\} + \lambda \| x \|^2$

3. **Online Matrix Completion:**
   • $K = \{X \in R^{n \times n} \mid \| X \|_* \leq k\}$ matrices with bounded nuclear norm
   • At time $t$, if $a_t = (i_t, j_t)$, then loss function $f_t(x) = (x(i_t, j_t) - b_t)^2$
Universal portfolio selection

Recall change in log-wealth:

$$\log W_{t+1} = \log W_t + \log (r_t^\top x_t)$$

Thus:

$$K = \Delta_n = \{ x \in \mathbb{R}^n , \sum_i x_i = 1 , x_i \geq 0 \}$$

$$f_t(x) = -\log(r_t^\top x_t)$$

A “Universal Portfolio” algorithm:

$$\frac{\text{Regret}}{T} = \frac{1}{T} \sum_t f_t(x_t) - \frac{1}{T} \min_{x^*} \sum_t f_t(x^*) \rightarrow 0$$
Bandit Online Convex Optimization

• Same as OCO, except $f_t$ is not revealed to learner, only $f_t(x_t)$ is.

• E.g. Ad selection: a bandit version of “prediction with expert advice”

• Decision set $K = \text{set of distributions over possible ads to be shown}$

\[ K = \Delta_n = \{ x \in \mathbb{R}^n , \sum_i x_i = 1 , x_i \geq 0 \} \]

• Loss function: let $c_t(i) = 0$ if ad $i$ would be clicked if it were shown, and $1$ otherwise

\[ f_t(x) = c_t^\top x = \sum_i c_t(i)x(i) \]
Algorithms
randomized weighted majority
follow-the-leader
online gradient descent, regularization?
log-regret, Online-Newton-Step
Bandit algorithms (FKM, barriers, volumetric spanners)
Projection-free methods (FW algorithm, online FW)

Part 1: Foundations
Prediction from expert advice

• $N$ “experts” (different regularizations)

• Can we perform as good as the best expert? (non-stochastic, adversarial setting)
Weighted Majority [Littlestone-Warmuth ‘89]

Binary classification: \( \{0, 1\} \) loss

Initially, all \( x_t(h_1) = 1 \)

\[
x_t(h_1) \quad x_t(h_2) \quad \ldots \quad x_t(h_n)
\]

round \( t \), predict by weighted majority

Update weights:

\[
x_{t+1}(h) = (1 - \eta)x_t(h) \quad \text{if expert } h \text{ errs}
\]

\[
x_{t+1}(h) = x_t(h) \quad \text{otherwise}
\]

Thm: Total-errors \( \leq 2(1 + \eta) \) Best-Expert + \( \frac{2\log(n)}{\eta} \)
Weighted Majority: Analysis

- **Total weight:** \( \Phi_t = \sum_{h \in H} x_t(h) \)

1. **Alg mistake \( \rightarrow \) at least half weight on experts who erred**
   \[
   \Phi_{t+1} \leq \frac{1}{2} \Phi_t (1 - \eta) + \frac{1}{2} \Phi_t \leq \Phi_t (1 - \frac{\eta}{2})
   \]

2. **If \( h^* \) is best expert,**
   \[
   \Phi_T \geq x_T(h^*) = (1 - \eta) \#(h^* \text{ errors})
   \]

- **Thus:**
   \[
   (1 - \eta) \#(\text{best expert errors}) \leq \Phi_T \leq \Phi_0 \cdot (1 - \frac{\eta}{2}) \#(\text{alg errors})
   \]

   \[
   \#(\text{alg errors}) \leq 2(1 + \eta) \#(\text{best expert errors}) + \frac{2 \log n}{\eta}
   \]
Randomized Weighted Majority [LW‘89]

general: expert \( h \) time \( t \): loss = \( c_t(h) \in [0,1] \)

\( n \) experts

Initially, all \( x_t(h_1) = 1 \)

\( x_t(h_1) \)

\( x_t(h_2) \) round \( t \), predict \( h \) w.p.: \( \frac{x_t(h)}{\sum_i x_t(h_i)} \)

\( \cdot \)

\( \cdot \) Update weights:

\( x_t(h_n) \) \( x_{t+1}(h) = e^{-\eta c_t(h)} \cdot x_t(h) \)

Thm: For \( \eta = \sqrt{\frac{\log N}{T}} \) Regret = \( O(\sqrt{T \log N}) \)
Reminder: Online Convex Optimization

Adversary

Convex cost function $f_t$

Online Player

Point $x_t$ in convex set $K$ in $\mathbb{R}^n$

Access to $f_t$?

Benchmark?

Regret = \[
\sum_t f_t(x_t) - \min_{x^* \in K} \sum_t f_t(x^*)
\]
Minimize regret: best-in-hindsight

\[
\text{Regret} = \sum_t f_t(x_t) - \min_{x^* \in K} \sum_t f_t(x^*)
\]

- Most natural:
  \[x_t = \arg \min \{ \sum_{i=1}^{t-1} f_i(x) \}\]

- Provably works: (even for non-convex sets [Kalai-Vempala’05])
  \[x_{t+1} = \arg \min_{x \in K} \left\{ \sum_{i=1}^t f_i(x) \right\}\]

- Is \(x_t\) close to \(x_{t+1}\)?

- Decision may be unstable! (teaser: this can be fixed w. regularization)
  \[x_t = \arg \min \left\{ \sum_{i=1}^t f_i(x) + \frac{1}{\eta} R(x) \right\}\]
Offline: Steepest Descent

- Move in the direction of steepest descent, which is:

\[-[\nabla f(x)]_i = -\frac{\partial}{\partial x_i} f(x)\]
Online gradient descent [Zinkevich ‘03]

\[ y_{t+1} = x_t - \eta \nabla f_t(x_t) \]

\[ x_{t+1} = \arg \min_{x \in K} \| y_{t+1} - x \| \]

Theorem: Regret = \( O(\sqrt{T}) \)
Analysis

Observation 1:

\[ \| y_{t+1} - x^* \|^2 = \| x_t - x^* \|^2 - 2\eta \nabla_t (x^* - x_t) + \eta^2 \| \nabla_t \|^2 \]

Observation 2: (Pythagoras)

\[ \| x_{t+1} - x^* \| \leq \| y_{t+1} - x^* \| \]

Thus:

\[ \| x_{t+1} - x^* \|^2 \leq \| x_t - x^* \|^2 - 2\eta \nabla_t (x^* - x_t) + \eta^2 \| \nabla_t \|^2 \]

Convexity:

\[
\sum_t \left[ f_t(x_t) - f_t(x^*) \right] \leq \sum_t \nabla_t (x_t - x^*) \\
\leq \frac{1}{\eta} \left( \| x_t - x^* \|^2 - \| x_{t+1} - x^* \|^2 \right) + \eta \sum_t \| \nabla_t \|^2 \\
\leq \frac{1}{\eta} \| x_1 - x^* \|^2 + \eta TG = O(\sqrt{T})
\]
Lower bound

\[ \text{Regret} = \Omega(\sqrt{T}) \]

- 2 experts, \( T \) iterations:
  - First expert has random loss in \{-1,1\}
  - Second expert loss = first * -1
- Expected loss = 0 (any algorithm)
- Regret = (think of first expert only)

\[ E[|\#1's - \#(-1)'s|] = \Omega(\sqrt{T}) \]
Algorithms

- randomized weighted majority
- follow-the-leader
- online gradient descent, regularization?
  log-regret, Online-Newton-Step
- Bandit algorithms (FKM, barriers, volumetric spanners)
- Projection-free methods (FW algorithm, online FW)

Part 2 coming up...