Tutorial: PART 2

Online Convex Optimization, A Game-Theoretic Approach to Learning

Elad Hazan
Princeton University

Satyen Kale
Yahoo Research
Exploiting curvature: logarithmic regret
Logarithmic regret

• Some natural problems admit better regret:
  • Least squares linear regression
  • Soft-margin SVM
  • Portfolio selection

• $\alpha$-strong convexity: (e.g. $f(x) = \| x \| ^2$)

\[ \nabla ^2 f(x) \succeq \alpha I \]

• $\alpha$-exp-concavity: (e.g. $f(x) = -\log a^T x$)

\[ \nabla ^2 f(x) \succeq \alpha \nabla f(x) \nabla f(x) ^T \]
Online Soft-Margin SVM

• Features $a \in \mathbb{R}^n$, labels $b \in \{-1, 1\}$

• Hypothesis class $H$ is a set of linear predictors:
  • Parameters: weight vectors $x \in \mathbb{R}^n$
  • Prediction of weight vector $x$ for feature vector $a$ is $a^T x$

• Loss $f_t(x) = \max\{0, 1 - b_t a_t^T x\} + \lambda \| x \|^2$
  Convex, non-smooth

  strongly-convex

• Thm [Hazan, Agarwal, Kale 2007]: OGD with step sizes $\eta_t \approx \frac{1}{t}$ has $O(\log T)$ regret
Universal Portfolio Selection

• Loss function $f_t(x) = -\log(r_t^T x)$

• Convex, but no strongly convex component

• But exp-concave: $\exp(-f_t(x)) = r_t^T x$ (concave function: actually, linear)

• Definition: $f$ is called $\alpha$-exp-concave if $\exp(-\alpha f(x))$ is concave

• Theorem [Hazan, Agarwal, Kale 2007]: For exp-concave losses, there is an algorithm, Online Newton Step, that obtains $O(\log T)$ regret
Online Newton Step algorithm

- Use $x_1$ = arbitrary point in $K$ in round 1
- For $t > 1$: use $x_t$ defined by following equations

\[
\begin{align*}
\nabla_{t-1} &= \nabla f_{t-1}(x_{t-1}) \\
C_{t-1} &= \sum_{\tau=1}^{t-1} \nabla_{\tau} \nabla_{\tau}^T \\
g_{t-1} &= \sum_{\tau=1}^{t-1} \nabla_{\tau} \nabla_{\tau}^T x_{\tau} - c\nabla_{\tau} \\
y_t &= C_{t-1}^{-1}g_{t-1} \\
x_t &= \arg\min_{x \in K} (x - y_t)^T C_{t-1} (x - y_t)^T
\end{align*}
\]

$\approx$ cumulative gradient step

$\approx$ cumulative Hessian

$c$ is a constant depending on $\alpha$, other bounds
Regularization
follow the regularized leader, online mirror descent
Follow-the-leader

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

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

- Be-the-leader algorithm has zero regret [Kalai-Vempala, 2005]:
  \[
x'_t = \arg\min_{x \in K} \sum_{i=1}^{t} f_i(x) = x_{t+1}
\]

- So if \( f_t(x_t) \approx f_t(x_{t+1}) \), we get a regret bound

- Unstability: \( f_t(x_t) - f_t(x_{t+1}) \) can be large!
Fixing FTL: Follow-The-Regularized-Leader (FTRL)

• First, suffices to use replace $f_t$ by a linear function, $\nabla f_t(x_t)^T x$

• Next, add regularization:

$$x_t = \arg \min_{x \in K} \sum_{i=1}^{t-1} \nabla f_i(x_i)^T x + \frac{1}{\eta} R(x)$$

• $R(x)$ is a strongly convex function, $\eta$ is a regularization constant

• Under certain conditions, this ensures stability:

$$\nabla f_t(x_t) \cdot x_t - \nabla f_t(x_t) \cdot x_{t+1} \approx O(\eta)$$
Specific instances of FTRL

• $R(x) = \| x \|^2$

\[
x_t = \arg\min_{x \in K} \sum_{i=1}^{t-1} \nabla f_i(x_i) \top x + \frac{1}{\eta} R(x)
\]

\[
= \Pi_K \left( -\eta \sum_{i=1}^{t-1} \nabla f_i(x_i) \right)
\]

• Here, $\Pi_K(y) = \arg\min_{x \in K} \| y - x \|$

• *Almost* like OGD: starting with $y_1 = 0$, for $t = 1, 2, ...$

\[
x_t = \Pi_K(y_t)
\]

\[
y_{t+1} = y_t - \eta \nabla f_t(x_t)
\]
Specific instances of FTRL

- Experts setting: \( K = \Delta_n \) distributions over experts
- \( f_t(x) = c^T_t x \), where \( c_t \) is the vector of losses
- \( R(x) = \sum_i x_i \log x_i \): negative entropy

\[
x_t = \arg \min_{x \in K} \sum_{i=1}^{t-1} \nabla f_i(x_i)^T x + \frac{1}{\eta} R(x)
\]

\[
= \exp \left( -\eta \sum_{i=1}^{t-1} c_i \right) / Z_t
\]

- *Exactly* RWM! Starting with \( y_1 = \) uniform distribution, for \( t = 1, 2, \ldots \)

\[
x_t = \prod_K (y_t)
\]

\[
y_{t+1} = y_t \odot \exp(-\eta c_t)
\]
FTRL ⇔ Online Mirror Descent

\[ x_t = \arg \min_{x \in K} \sum_{i=1}^{t-1} \nabla f_i(x_i) \top x + \frac{1}{\eta} R(x) \]

**Bregman Projection:**

\[ \Pi^K_R(y) = \arg \min_{x \in K} B_R(x \| y) \]

\[ B_R(x \| y) := R(x) - R(y) - \nabla R(y) \top (x - y) \]

\[ x_t = \Pi^K_R(y_t) \]

\[ y_{t+1} = (\nabla R)^{-1}(\nabla R(y_t) - \eta \nabla f_t(x_t)) \]
Advanced regularization follow the perturbed leader, AdaGrad
Randomized regularization: Follow-The-Perturbed-Leader

• Special case of OCO, Online Linear Optimization: \( f_t(x) = c_t^T x \)

• Follow-The-Perturbed-Leader [Kalai-Vempala 2005]:

\[
x_t = \arg \min_{x \in K} \sum_{i=1}^{t-1} c_i^T x + p^T x
\]

• Regularization \( p^T x \): vector \( p \) drawn u.a.r. from \([-\sqrt{T}, \sqrt{T}]\)

• Expected regret is the same if \( p \) is chosen afresh every round

• Re-randomizing every round \( \Rightarrow x_t = x_{t+1} \) w.p. at least \( 1 - 1/\sqrt{T} \)

\( \Rightarrow \) regret = \( O(\sqrt{T}) \)
Follow-The-Perturbed-Leader

• Algorithm: $p$ is random vector with entries in $[-\sqrt{T}, \sqrt{T}]$

$$x_t = \arg \min_{x \in K} \sum_{i=1}^{t-1} c_i^T x + p^T x$$

• Implementation requires only linear optimization over $K$, easier than projections needed by OMD

• $O(\sqrt{T})$ regret bound holds for arbitrary sets $K$: convexity unnecessary!

• Can be extended to convex losses using gradient trick + using expected perturbed leader in each round
Adaptive Regularization: AdaGrad

• Consider use of OGD in learning a generalized linear model
• For parameter $x \in \mathbb{R}^n$ and example $(a, b) \in \mathbb{R}^n \times \mathbb{R}$, prediction is some function of $a^T x$
• Thus, $\nabla f_t(x) = \ell(a_t, b_t, x)a_t$

• OGD update: $x_{t+1} = x_t - \eta \nabla f_t(x_t) = x_t - \eta \ell(a_t, b_t, x)a_t$
• Thus, features treated equally in updating parameter vector

• In typical text classification tasks, feature vectors $a_t$ are very sparse
• Slow learning!

• Adaptive regularization: implements per-feature learning rates
AdaGrad (diagonal form)
[Duchi, Hazan, Singer ’10]

• Set $x_1 \in K$ arbitrarily
• For $t = 1, 2,...$
  1. use $x_t$ obtain $f_t$
  2. compute $x_{t+1}$ as follows:

$$G_t = \text{diag}(\sum_{i=1}^t \nabla f_i(x_i)\nabla f_i(x_i)^\top)$$

$$y_{t+1} = x_t - \eta G_t^{-1/2}\nabla f_t(x_t)$$

$$x_{t+1} = \arg \min_{x \in K} (y_{t+1} - x)^\top G_t(y_{t+1} - x)$$

• Regret bound: $O\left(\sum_i \sqrt{\sum_t \nabla f_t(x_t)_i^2}\right)$

• Infrequently occurring, or small-scale, features have small influence on regret (and therefore, convergence to optimal parameter)
Bandit Convex Optimization
Bandit Optimization

• Motivating example: ad selection on websites

• Simplest form: in each round $t$
  • Choose one ad $i_t$ out of $n$
  • Simultaneously, user chooses loss vector $c_t$
  • Observe loss of chosen ad, $c_t(i_t)$

\[
\text{Regret} = \sum_{t=1}^{T} c_t(i_t) - \min_i \sum_{t=1}^{T} c_t(i)
\]

• Essentially, experts problem with partial feedback
• Modeled as online linear optimization over $n$-simplex of distributions
• Available choices are usually called “arms” (from multi-armed bandits)
• Randomization necessary, so consider expected regret

\[c_t(i) = -\text{revenue if ad } i \text{ would be clicked if shown, 0 otherwise}\]
Constructing cost estimators

Exploration:
• Choose arm $i_t$ from 1, 2, ..., $n$ u.a.r.
• Observe $c_t(i_t)$

\[ \tilde{c}_t = n c_t(i_t) e_{i_t} \]

\[ \mathbb{E}[\tilde{c}_t] = c_t \]

Exploitation:
\[ x_t = \arg \min_{x \in K} \sum_{i=1}^{t-1} \tilde{c}_i \top x + \frac{1}{\eta} R(x) \]
Separate explore-exploit template

Algorithm:
1. w.p. $q$ explore (as before), create estimator $\mathbb{E}[\tilde{c}_t] = c_t$

2. o/w exploit

$$x_t = \arg \min_{x \in K} \sum_{i=1}^{t-1} \tilde{c}_i^\top x + \frac{1}{\eta} R(x)$$

Regret is essentially (ignoring dependence on $n$) bounded by

$$\approx q \cdot T + (1 - q) \cdot \frac{1}{q} \cdot \text{Regret}_{FTRL} = O \left( q + \frac{\sqrt{T}}{q} \right) = O(T^{2/3})$$

Can we do better?
Simultaneous exploration-exploitation
[Auer, Cesa-Bianchi, Freund, Schapire 2002]

Exploration+Exploitation:
• Choose arm $i_t$ from distribution $x_t$
• Observe $c_t(i_t)$

$\tilde{c}_t = \frac{c_t(i_t)}{x_t(i_t)} e_{i_t}$

$\mathbb{E}[\tilde{c}_t] = C_t$

$x_t = \arg\min_{x \in K} \sum_{i=1}^{t-1} \tilde{c}_i^\top x + \frac{1}{\eta} R(x)$

With $R = \text{negative entropy}$, this alg (called EXP3) has $O(\sqrt{T})$ regret
Bandit Linear Optimization

- Generalize from multi-armed bandits
- $K$: convex set in $\mathbb{R}^n$
- Linear cost functions: $f_t(x) = c_t^T x$

- Approach:
  1. (Exploration) construct unbiased estimators $\mathbb{E}[\tilde{c}_t] = c_t$
  2. (Exploitation) Plug into FTRL:

$$x_t = \arg\min_{x \in K} \sum_{i=1}^{t-1} \tilde{c}_i^\top x + \frac{1}{\eta} R(x)$$
Bandit Linear Optimization

• $K$: convex set in $\mathbb{R}^n$, linear cost functions: $f_t(x) = c_t^T x$

• Assume $K$ contains $e_1, e_2, ..., e_n$ “exploration basis”

• In round $t$:
  $$x_t = \arg\min_{x \in K} \sum_{i=1}^{t-1} \tilde{c}_i \top x + \frac{1}{\eta} R(x)$$

1. W.p. $q$ choose $i_t \in \{1, 2, ..., n\}$ and use $e_{i_t}$. Observe $c_t(i_t)$ and set
   $$\tilde{c}_t = \frac{c_t(i_t)}{q/n} e_{i_t}$$

2. W.p. $1-q$ use $x_t$ obtained from FTRL and set $\tilde{c}_t = 0$
   $$\mathbb{E}[\tilde{c}_t] = c_t$$

• Gives regret bound of $O(T^{2/3})$
Geometric regularization via barriers
[Abernethy, Hazan, Rakhlin 2008]

• $R =$ self-concordant barrier for $K$.

• **Dikin ellipsoid** at any point $x$ lies in $K$:
  $\{y| (y - x)^T \nabla^2 R(x)(y - x) \leq 1\}$

• **Algorithm**: use random endpoint of axes of ellipsoid at $x_t$
  • $\equiv$ using $x_t$ in expectation (exploitation)
  • $2n$ endpoints suffice to construct $\tilde{C}_t$ (exploration)

• Thm: Regret $= O(\sqrt{T})$

$$x_t = \arg \min_{x \in K} \sum_{i=1}^{t-1} \tilde{c}_i^T x + \frac{1}{\eta} R(x)$$
Bandit Online Convex Optimization
[Flaxman, Kalai, McMahan 2005]

$f_t$ arbitrary convex functions

Approach:
1. Construct estimator $\tilde{c}_t$ of $\nabla f_t(x_t)$
2. Plug into FTRL

$$x_t = \arg\min_{x \in K} \sum_{i=1}^{t-1} \tilde{c}_i^\top x + \frac{1}{\eta} R(x)$$

$$\nabla f(x) \approx \frac{1}{\delta} \mathbb{E}_{u \in S^{n-1}} \left[ f(x + \delta u) u \right]$$

- Near boundary, estimators have large variance
- Hence, $x_t$ and $x_{t+1}$ can be very different
- Regret bound = $O(T^{3/4})$
- Recent developments: $O(\sqrt{T})$ regret possible!
Projection-free algorithms
Matrix completion

In round $t$:
- Obtain (user, movie) pair, and output a recommended rating
- Then observe true rating, and suffer square loss

Comparison class: all matrices with bounded trace norm
Bounded trace norm matrices

- Trace norm of a matrix = sum of singular values
- \( K = \{ X \mid X \text{ is a matrix with trace norm at most } D \} \)
- Optimizing over \( K \) promotes low rank solutions

- Online Matrix Completion problem is an OCO problem over \( K \)
- FTRL, OMD etc require computing projections on \( K \)
- Expensive! Requires computing eigendecomposition of matrix to be projected: \( O(n^3) \) operation

- But: linear optimization over \( K \) is easier
- Requires computing top eigenvector; can be done in \( O(\text{sparsity}) \) time typically
1. Matrix completion ($K = \text{bounded trace norm matrices}$) 
   - eigen decomposition
   - largest singular vector computation

2. Online routing ($K = \text{flow polytope}$) 
   - conic optimization over flow polytope
   - shortest path computation

3. Rotations ($K = \text{rotation matrices}$) 
   - conic optimization over rotations set
   - Wahba’s algorithm – linear time

4. Matroids ($K = \text{matroid polytope}$) 
   - convex opt. via ellipsoid method
   - greedy algorithm – nearly linear time!
Projection-Free Online Learning

• Is online learning possible given access to linear optimization oracle rather than projection oracle?

• Yes: projection can be implemented via polynomially many linear optimizations

• But for efficiency, would like only a few (1 or 2) calls to linear optimization oracle per round

• Follow-The-Perturbed-Leader does the job for linear loss functions

• What about general convex loss functions? Is OCO possible with only 1 or 2 calls to a linear optimization oracle per round?
Conditional Gradient algorithm [Frank, Wolfe ’56]

Convex opt problem:
\[
\min_{x \in K} f(x)
\]

Assume:
- \( f \) is smooth, convex
- linear opt over \( K \) is easy

\[
\begin{align*}
  v_t &= \arg \min_{x \in K} \nabla f(x_t)^\top x \\
  x_{t+1} &= x_t + \eta_t (v_t - x_t)
\end{align*}
\]

1. At iteration \( t \): convex comb. of at most \( t \) vertices \((\text{sparsity})\)
2. No learning rate. \( \eta_t \approx \frac{1}{t} \) (independent of diameter, gradients etc.)
Online Conditional Gradient [Hazan, Kale ’12]

- Set $x_1 \in K$ arbitrarily
- For $t = 1, 2, \ldots$,
  1. Use $x_t$, obtain $f_t$
  2. Compute $x_{t+1}$ as follows

\[
\begin{align*}
  v_t &= \arg \min_{x \in K} \left( \sum_{i=1}^{t} \nabla f_i(x_i) + \beta_t x_t \right)^T x \\
  x_{t+1} &\leftarrow (1 - t^{-\alpha})x_t + t^{-\alpha} v_t
\end{align*}
\]

\[\sum_{i=1}^{t} \nabla f_i(x_i) + \beta_t x_t\]

Theorem: Regret $= O(T^{3/4})$
Theorem: For polytopes, strongly-convex and smooth losses:
1. Offline: gap after $t$ steps: $\exp(-ct)$
2. Online: $\text{Regret} = O(\sqrt{T})$
Directions for future research

1. Efficient algorithms & optimal dependency on dimension for bandit OCO
2. Faster online matrix algorithms
3. Optimal regret /running time tradeoffs
4. Optimal regret and projection-free algorithm for general convex sets
5. Time series/dependence/adaptivity (vs. static regret)
6. Adding state / online Reinforcement Learning / MDPs / stochastic games [lots of existing work...]
7. Tensor/spectral learning in the online setting
8. Game theoretic aspect – stronger notions, approachability, ...
Bibliography & more information, see:


Thank you!