Lecture 16:
Mirror Descent
Online to Batch, Stochastic Optimization
Stability and Learning

\(\ell(w, z)\) is \(G\)-Lipschitz w.r.t \(\|w\|\)

\[\hat{w}_\lambda = \min L_S(w) + \lambda \Psi(w)\] is \(\beta(m) = \frac{G^2}{\lambda \alpha m}\) stable

\(\frac{G^2}{\lambda \alpha m}\) regret on \(\tilde{\ell}_\lambda (w, z) = \ell(w, z) + \lambda \Psi(w)\)

\[\sqrt{\frac{B^2 G^2}{\alpha m}}\] regret on \(\ell(w, z)\) over \(\mathcal{H} = \{\Psi(w) \leq B^2\}\) with FTRL

\[L(\hat{w}_\lambda) \leq \inf_{w} \tilde{L}_\lambda (w) + \frac{G^2}{\lambda \alpha m}\]

\[L(\hat{w}_\lambda) \leq \inf_{\Psi(w) \leq B^2} L(w) + \sqrt{\frac{B^2 G^2}{\alpha m}}\]

\(\frac{B^2 G^2}{\alpha m}\) regret on \(\ell(w, z)\) over \(\mathcal{H} = \{\Psi(w) \leq B^2\}\) with L-FTRL (dual averaging)
Linearized FTRL

\[ w_{t+1} = \arg \min_{w \in \mathcal{H}} \frac{1}{t} \sum_{i=1}^{t} \langle \nabla z_i(w_i), w \rangle + \lambda_t \Psi(w) \]

- If \( \overline{\mathcal{H}} = \mathcal{B} \):
  \[ w_{t+1} = \nabla \Psi^{-1} \left( -\frac{1}{\lambda_t} \sum_{i=1}^{t} \nabla z_i(w_i) \right) \]
  \[ = \nabla \Psi^{-1} \left( \frac{\lambda_{t-1}(t-1)}{\lambda_t t} \nabla \Psi(w_t) - \frac{1}{\lambda_t} \nabla z_i(w_i) \right) \]

- If \( \overline{\mathcal{H}} \subset \mathcal{B} \):
  \[ w_{t+1} = \Pi_{\mathcal{H}^*} \left( \nabla \Psi^{-1} \left( -\frac{1}{\lambda_t} \sum_{i=1}^{t} \nabla z_i(w_i) \right) \right) \]
Linearized FTRL

\[ w_{t+1} = \arg \min_{w \in \mathcal{H}} \frac{1}{t} \sum_{i=1}^{t} \langle \nabla z_i(w_i), w \rangle + \lambda_t \Psi(w) \]

- If \( \mathcal{H} = \mathcal{B} \):
  \[ w_{t+1} = \nabla \Psi^{-1} \left( -\frac{1}{\lambda_t t} \sum_{i=1}^{t} \nabla z_i(w_i) \right) \]
  \[ = \nabla \Psi^{-1} \left( \frac{\lambda_{t-1}(t-1)}{\lambda_t t} \nabla \Psi(w_t) - \frac{1}{\lambda_t t} \nabla z_i(w_i) \right) \]

- If \( \mathcal{H} \subset \mathcal{B} \):
  \[ w_{t+1} = \Pi_{\Psi}^{\mathcal{H}} \left( \nabla \Psi^{-1} \left( -\frac{1}{\lambda_t t} \sum_{i=1}^{t} \nabla z_i(w_i) \right) \right) \]

\( \mathcal{B} \)
\( \mathcal{H} \)
\( \mathcal{B}^* \)
From FTRL to Mirror Descent

• FTRL easily implementable only for linear objectives (or for a linearization)---**otherwise need to store all the history**

• Linearized FTRL *almost* yields online gradient descent:

\[
\begin{align*}
    w_{t+1} &= \frac{\lambda_{t-1}(t - 1)}{\lambda_t t} w_t - \frac{1}{2\lambda_t t} \nabla z_t(w_t)
\end{align*}
\]

*but only if $\mathcal{H} = \mathcal{B}$*
Non-Linearized Mirror Descent

Initialize: \( w_t = \arg \min_{w \in \mathcal{H}} \Psi(w) \)
Iterate: \( w_{t+1} = \arg \min_{w \in \mathcal{H}} \ell(w, z_t) + t\lambda_tD_\Psi(w||w_t) \)

- Enjoys same regret guarantees as Follow-The-Regularized-Leader:
- For a convex \( G \)-Lipschitz problem w.r.t. norm \( \|w\| \), if \( \Psi(w) \geq 0 \) is \( \alpha \)-strongly convex w.r.t. \( \|w\| \), for any \( w \in \mathcal{H} \),
  \[
  \frac{1}{m} \sum_{t=1}^{m} \ell(w_t, z_t) \leq \frac{1}{m} \sum_{t=1}^{m} \ell(w, z_t) + \frac{1}{m} \sum_{t=1}^{m} \lambda_t \Psi(w) + \frac{1}{m} \sum_{t=1}^{m} \frac{2G^2}{\alpha\lambda_t t} 
  \]
- If also \( \forall w \in \mathcal{H} \) \( \Psi(w) \leq B^2 \), using \( \lambda_t = \sqrt{\frac{2G^2}{\alpha B^2 t}} \), for any \( w \in \mathcal{H} \),
  \[
  \frac{1}{m} \sum_{t=1}^{m} \ell(w_t, z_t) \leq \frac{1}{m} \sum_{t=1}^{m} \ell(w, z_t) + \sqrt{\frac{32G^2B^2}{\alpha m}}
  \]
(Linearized) Mirror Descent

- **Linearized Mirror Descent:**
  \[
  w_{t+1} = \arg \min_{w \in \mathcal{H}} \langle \nabla z_t(w_t), w \rangle + t\lambda_t D_\Psi(w||w_t)
  \]
  \[
  = \Pi_{\overline{\mathcal{H}}_{\Psi}} \left( \nabla \Psi^{-1} \left( \nabla \Psi(w_t) - \frac{1}{t\lambda_t} \nabla z_t(w_t) \right) \right)
  \]

- **Contrast with Linearized FLTR (dual averaging):**
  \[
  w_{t+1} = \Pi_{\overline{\mathcal{H}}_{\Psi}} \left( \nabla \Psi^{-1} \left( -\frac{1}{\lambda_t} \sum_{i=1}^{t} \nabla z_i(w_i) \right) \right)
  \]
  \[
  = \nabla \Psi^{-1} \left( \frac{\lambda_t^{-1}(t-1)}{\lambda_t} \nabla \Psi(w_t) - \frac{1}{\lambda_t} \nabla z_i(w_i) \right)
  \]

- **Only if \( \overline{\mathcal{H}} = \mathcal{B} \)**
Online Gradient Descent

- With $Ψ(w) = \|w\|^2$, Online Mirror Descent becomes Online Projected Gradient Descent:

$$w_{t+1} = \arg\min_{w \in \mathcal{H}} \langle \nabla z_t(w_t), w \rangle + t\lambda_t \|w - w_t\|^2$$

$$= \Pi_{\mathcal{H}}(w_t - \eta_t \nabla z_t(w_t))$$

$\Pi_{\mathcal{H}}(w) = \arg\min_{w' \in \mathcal{H}} \|w - w'\|$
Advantages of Mirror Descent over FTRL / Dual Averaging

• Allows direct derivation of Online Gradient Descent with arbitrary stepsizes and without scaling of iterates

• Allows incremental algorithm (keeping only $w_t$, and not entire history) *even without linearization*

• Can also partially linearize, or use any other convex lower bound on the objective

• E.g., for: $\ell(w, (x, y)) = loss(\langle w, x \rangle; y) + \|w\|_1$ can use:
  • $w_{t+1} = \arg \min_{w \in \mathcal{H}} loss(\langle w, x \rangle; y) + \|w\|_1 + t\lambda_t D_\Psi(w \| w_t)$
  • $w_{t+1} = \arg \min_{w \in \mathcal{H}} \langle \nabla loss(w_t), w \rangle + \|w\|_1 + t\lambda_t D_\Psi(w \| w_t)$
Summary

• For a convex $G$-Lipschitz problem w.r.t. norm $\|w\|$, if we have a $\alpha$-strongly convex regularizer $0 \leq \Psi(w)$ s.t. $\forall w \in \mathcal{H} \Psi(w) \leq B^2$, we can get regret $O\left(\sqrt{\frac{B^2G^2}{\alpha m}}\right)$ using:
  - FTRL
  - Linearized FLTR (dual averaging)
  - Non-Linearized MD
  - Linearized MD (e.g. Projected GD)
  - Partially linearized MD

• Under same condition, using stability arguments, RERM agnostically PAC learns with sample complexity $O\left(\frac{B^2G^2}{\alpha \epsilon^2}\right)$ in the statistical setting

• And... it's also possible to show matching bounds on the Radamacher complexity of norm-constrained linear predictors is such a strongly convex regularizer exists
Online vs Statistical Learning

• Problem specified by:
  • Domain and objective: $\ell: \overline{\mathcal{H}} \times \mathcal{Z} \rightarrow \mathbb{R}$
  • Hypothesis class to compete with: $\mathcal{H} \subseteq \overline{\mathcal{H}}$

• Online Regret of learning rule $A: \mathcal{Z}^* \rightarrow \overline{\mathcal{H}}$

$$\forall z_1, z_2, \ldots, z_m \frac{1}{m} \sum_{t=1}^{m} \ell(A(z_1 \ldots z_{t-1}), z_t) \leq \inf_{h \in \mathcal{H}} \frac{1}{m} \sum_{t=1}^{m} \ell(h, z_t) + \text{Reg}(m)$$

• Statistical (PAC) “Regret” / excess error:

$$\forall D(\mathcal{Z}) \forall \delta \sim D \inf_{S \sim D} \frac{1}{m} \sum_{t=1}^{m} L_D(A(S)) \leq \inf_{h \in \mathcal{H}} \frac{1}{m} \sum_{t=1}^{m} L_D(h) + \epsilon(m, \delta)$$

Low Statistical Regret $\not\Rightarrow$ Low Online Regret
(e.g. linear classification)

• But does low online regret imply low statistical regret?
Online ⇒ Batch ?

• If $A: \mathcal{Z}^* \to \overline{\mathcal{H}}$ has low online regret $\text{Reg}(m)$, can we use $A(S)$ for statistical learning and guarantee low excess error?

• Example: $\forall z \ell(h_{\text{good}}, z) = 0, \ell(h_{\text{bad}}, z) = 1$

\[
A(z_1, ..., z_t) = \begin{cases} 
  h_{\text{good}}, & t \neq 1000 \\
  h_{\text{bad}}, & t = 1000 
\end{cases}
\]

$\text{Reg}(1000) = 0.001$, but for $m = 1000$, $L(A(S)) = \inf_h L(h) + 1$

• Solution: return an average of the all online iterates.
Online ⇒ Batch

• For a convex learning problem (i.e. \( \mathcal{H} \) is convex and \( \ell(h, z) \) is convex in \( h \)):

\[
\overline{A}(z_1, ..., z_m) = \frac{1}{m} \sum_{t=1}^{m} A(z_1, ..., z_{t-1})
\]

\( \overline{A}(S) \)- Batch Conversion of online rule \( A \)

Input: training set \( S = \{z_1, z_2, ..., z_m\} \)

1. Run \( A \) on \( z_1, ..., z_m \) to obtain \( h_1, h_2, ..., h_{m+1} \)

2. Return \( \overline{h}_m = \frac{1}{m} \sum_{t=1}^{m} h_t \)

• Theorem: \( \mathbb{E}_{S \sim \mathcal{D}^m} \left[ L_{\mathcal{D}} \left( \overline{A}(S) \right) \right] \leq \inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + Reg(m) \)

Proof: \( \mathbb{E} \left[ L \left( \overline{A}(S) \right) \right] = \mathbb{E} \left[ L \left( \frac{1}{m} \sum_{t=1}^{m} h_t \right) \right] \leq \mathbb{E} \left[ \frac{1}{m} \sum_{t=1}^{m} L(h_t) \right] = \mathbb{E} \left[ \frac{1}{m} \sum_{t=1}^{m} \ell(h_t, z_t) \right] \leq \mathbb{E} \left[ \frac{1}{m} \sum_{t=1}^{m} \ell(h, z_t) + Reg(m) \right] = L(h) + Reg(m) \)

• High probability?

If \( |\ell(h, z, )| \leq a, \quad \forall_{S \sim \mathcal{D}^m} L_{\mathcal{D}} \left( \overline{A}(S) \right) \leq \inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + Reg(m) + 3a \sqrt{\frac{\log 1/\delta}{m}} \)

Proof sketch: apply Azuma’s inequality (generalization of Hoeffding)
Non-Convex: Online $\Rightarrow$ Batch

- If learning problem is non-convex?
  - E.g. supervised learning problem with 0/1 loss.
- Return randomized predictor:
  \[
  \overline{A}(S) = h_t \text{ with probability } \frac{1}{m}
  \]
- Guarantee is now on $\tilde{L}_D (\overline{A}(S)) \overset{\text{def}}{=} \mathbb{E}_{\text{rand}} \left[ L_D (\overline{A}(S)) \right]$
- Can be viewed as convexifying the problem:
  \[
  \overline{\mathcal{H}} = \text{conv}(\mathcal{H}) \approx \{ P(\mathcal{H}) \mid P \text{ is a dist over } \mathcal{H} \}
  \]
  \[
  \tilde{\ell} (\sum_i \alpha_i h_i, z) = \sum_i \alpha_i \ell (h_i, z)
  \]
  Or: $\tilde{\ell} (P, z) = \mathbb{E}_{h \sim P} [\ell (h, z)]$
Implications of Online-to-Batch

• Anything online learnable is also statistically learnable, with same regret/excess error
  • Explains why we got same expressions, eg for finite hypothesis classes, large margin linear learning, $\ell_p$-regularized problems
  • Converse not true!
  • Important: online learnable $\not\Rightarrow$ ULLN nor learnability with ERM.
    • E.g.: generalized convex Lipschitz problems like mean estimation with missing data
    • Learnable with online-to-batch, but not ERM
    • Also learnable with RERM (established using stability)

• More computationally efficient learning methods
  • E.g. Perceptron(+online-to-batch) vs SVM

• Useful even as optimization approach!
Optimize $F(w) = \mathbb{E}_{z \sim \mathcal{D}}[f(w, z)]$ s.t. $w \in \mathcal{W}$

1. Initialize $w_1 = 0 \in \mathcal{W}$
2. At iteration $t = 1, 2, 3, \ldots$
   1. Sample $z_t \sim \mathcal{D}$
   2. $w_{t+1} = \Pi_\mathcal{W} \left( w_t - \eta_t \nabla f(w_t, z_t) \right)$
3. Return $\overline{w}_m = \frac{1}{m} \sum_{t=1}^{m} w_t$

If $\|\nabla f(w, z)\|_2 \leq G$ then with appropriate step size:

$$
\mathbb{E}[F(\overline{w}_m)] \leq \inf_{w \in \mathcal{W}, \|w\|_2 \leq B} F(w) + O \left( \sqrt{\frac{B^2 G^2}{m}} \right)
$$

Similarly, also Stochastic Mirror Descent
Stochastic Optimization

\[ \min_{w \in \mathcal{W}} F(w) = \mathbb{E}_{z \sim \mathcal{D}}[f(w, z)] \]

Based only on stochastic information on \( F \)

- Only unbiased estimates of \( F(w), \nabla F(w) \)
- No direct access to \( F \)

E.g., fixed \( f(w, z) \) but \( \mathcal{D} \) unknown

- Optimize \( F(w) \) based on iid sample \( z_1, z_2, ..., z_m \sim \mathcal{D} \)
- \( g = \nabla f(w, z_t) \) is unbiased estimate of \( \nabla F(w) \)

- Traditional applications
  - Optimization under uncertainty
    - Uncertainty about network performance
    - Uncertainty about client demands
    - Uncertainty about system behavior in control problems
  - Complex systems where it’s easier to sample then integrate over \( z \)
Machine Learning is Stochastic Optimization

$$\min_h \mathbb{E}_{z \sim \mathcal{D}}[\ell(h, z)] = \mathbb{E}_{x, y \sim \mathcal{D}}[\text{loss}(h(x), y)]$$

- Optimization variable: predictor $h$
- Objective: generalization error $L(h)$
- Stochasticity over $z = (x, y)$

“General Learning” ≡ Stochastic Optimization:
Stochastic Optimization vs Statistical Learning

Focus on computational efficiency
- Generally assumes unlimited sampling
  - as in monte-carlo methods for complicated objectives
- Optimization variable generally a vector in a normed space
  - complexity control through norm
- Mostly convex objectives

Focus on sample size
- What can be done with a fixed number of samples?
- Abstract hypothesis classes
  - linear predictors, but also combinatorial hypothesis classes
  - generic measures of complexity such as VC-dim, fat shattering, Radamacher
- Also non-convex classes and loss functions

Arkadi Nemirovskii

Valdimir Vapnik
Two Approaches to Stochastic Optimization / Learning

$$\min_{w \in \mathcal{W}} F(w) = \mathbb{E}_{z \sim \mathcal{D}}[f(w, z)]$$

- **Empirical Risk Minimization (ERM) / Sample Average Approximation (SAA):**
  - Collect sample $z_1, \ldots, z_m$
  - Minimize $F_S(w) = \frac{1}{m} \sum_i f(w, z_i)$
  - Analysis typically based on Uniform Convergence

- **Stochastic Approximation (SA):** [Robins Monro 1951]
  - Update $w_t$ based on $z_t$
    - E.g., based on $g_t = \nabla f(w, z_t)$
  - E.g.: stochastic gradient descent
  - Online-to-batch conversion of online algorithm...
SGD as an Optimization Algorithm

Goal \( \min_{\|w\|_2 \leq B} F(w) \)

- If \( F(w) \) is \( G \)-Lipschitz (and all gradient estimates are bounded by \( G \)), after \( T = O \left( \frac{B^2 G^2}{\epsilon^2} \right) \) iterations of SGD, we have
  \[
  F(\overline{w}_T) \leq \inf_{\|w\|_2 \leq B} F(w) + \epsilon
  \]

- Claim: any optimization algorithm that only accesses \( F(w) \) locally, i.e. only accesses \( F(w), \nabla F(w), \) etc, and ensures \( F(\overline{w}_T) \leq \inf_{\|w\|_2 \leq B} F(w) + \epsilon \) after \( T \) accesses for any \( G \)-Lipschitz objective, must make at least \( T = \Omega \left( \frac{B^2 G^2}{\epsilon^2} \right) \) accesses

- SGD required only \( T = O \left( \frac{B^2 G^2}{\epsilon^2} \right) \) accesses to unbiased stochastic gradient estimates, i.e. \( g \) s.t. \( \mathbb{E}[g] \in \partial F(w) \)

- E.g., consider \( F(w) = L_S(w) = \frac{1}{m} \sum_{i=1}^{m} \text{loss}^\text{hinge}(\langle w, \phi(x_i) \rangle; y_i) \)
SGD for SVM

\[
\min L_S(w) \text{ s.t. } \|w\|_2 \leq B
\]

Use \( g_t = \nabla_w \text{loss}^{\text{hinge}}(\langle w_t, \phi_{i_t}(x) \rangle; y_{i_t}) \) for random \( i_t \)

Initialize \( w^{(0)} = 0 \)

At iteration \( t \):

- Pick \( i \in 1 \ldots m \) at random
- If \( y_i \langle w^{(t)}, \phi(x_i) \rangle < 1 \),
  \[
  w^{(t+1)} \leftarrow w^{(t)} + \eta_t y_i \phi(x_i)
  \]
  else: \( w^{(t+1)} \leftarrow w^{(t)} \)

- If \( \|w^{(t+1)}\|_2 > B \), then \( w^{(t+1)} \leftarrow B \frac{w^{(t+1)}}{\|w^{(t+1)}\|_2} \)

Return \( \overline{w}^{(T)} = \frac{1}{T} \sum_{t=1}^{T} w^{(t)} \)

\[
\|g_t\|_2 \leq G \Rightarrow L_S(\overline{w}^{(T)}) \leq L_S(\overline{w}) + \sqrt{\frac{B^2G^2}{T}}
\]

(in expectation over randomness in algorithm)
Stochastic vs Batch

\[
\min L_S(w) \ s.t. \ \|w\|_2 \leq B
\]

\[
\begin{align*}
g_1 &= \nabla \text{loss}(w \text{ on } (x_1, y_1)) \\
g_2 &= \nabla \text{loss}(w \text{ on } (x_2, y_2)) \\
g_3 &= \nabla \text{loss}(w \text{ on } (x_3, y_3)) \\
g_4 &= \nabla \text{loss}(w \text{ on } (x_4, y_4)) \\
g_5 &= \nabla \text{loss}(w \text{ on } (x_5, y_5)) \\
\vdots \\
g_m &= \nabla \text{loss}(w \text{ on } (x_m, y_m))
\end{align*}
\]

\[
\begin{array}{|c|c|}
\hline
x_1, y_1 & \hline
x_2, y_2 & \hline
x_3, y_3 & \hline
x_4, y_4 & \hline
x_5, y_5 & \hline
\vdots & \hline
x_m, y_m & \hline
\end{array}
\]

\[
\nabla L_S(w) = \frac{1}{m} \sum g_i
\]

\[
\begin{align*}
w &\leftarrow w - g_1 \\
w &\leftarrow w - g_2 \\
w &\leftarrow w - g_3 \\
w &\leftarrow w - g_4 \\
w &\leftarrow w - g_5 \\
w &\leftarrow w - g_{m-1} \\
w &\leftarrow w - g_m
\end{align*}
\]

\[
w &\leftarrow w - \sum g_i
\]
Stochastic vs Batch

• Intuitive argument: if only taking simple gradient steps, better to be stochastic

• To get $L_S(w) \leq L_S(\hat{w}) + \epsilon_{opt}$:

<table>
<thead>
<tr>
<th></th>
<th>#iter</th>
<th>cost/iter</th>
<th>runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch GD</td>
<td>$B^2 G^2 / \epsilon_{opt}^2$</td>
<td>$md$</td>
<td>$md \frac{B^2 G^2}{\epsilon_{opt}^2}$</td>
</tr>
<tr>
<td>SGD</td>
<td>$B^2 G^2 / \epsilon_{opt}^2$</td>
<td>$d$</td>
<td>$d \frac{B^2 G^2}{\epsilon_{opt}^2}$</td>
</tr>
</tbody>
</table>

• What about $L(w)$, which is what we really care about?
• How small should $\epsilon_{opt}$ be?
• Comparison to methods with a log $1/\epsilon$ dependence that use the structure of $L_S(w)$ (not only local access)?