Learning and Optimization: Lower Bounds and Tight Connections

Nati Srebro
TTI-Chicago

On The Universality of Online Mirror Descent
  S, Karthik Sridharan (UPenn), Ambuj Tewari (Michigan), NIPS’11
Learning from an Optimization Viewpoint
  Karthik Sridharan TTIC PhD Thesis
Learning/Optimization over $L_2$ Ball

- Stat Learning / Stoch Optimization:

$$\min_{\|w\|_2 \leq B} L(w) = \mathbb{E}_{x,y \sim D}[\ell(\langle w, x \rangle; y)]$$

based on $m$ iid samples $x, y \sim D$

- Using SAA/ERM: $\hat{w} = \arg \min \hat{L}(w)$

$$L(\hat{w}) \leq \inf_{\|w\| \leq B} L(w) + 2\sqrt{B^2 R^2 / m}$$

- Rate of 1st order (or any local) optimization:

$$\hat{\mathcal{L}}(w_T) \leq \inf_{\|w\| \leq B} \hat{\mathcal{L}}(w) + \sqrt{B^2 R^2 / T}$$

- Using SA/SGD on $L(w)$: $w_{t+1} \leftarrow w_t - \eta_t \nabla_w \ell(\langle w, x_t \rangle; y_t)$

$$L(\bar{w}_m) \leq \inf_{\|w\| \leq B} L(w) + \sqrt{B^2 R^2 / m}$$

[Bottou Bousquet 08][Shalev-Shwartz 08][Juditsky Lan Nemirovski Shapiro 09]
Learning/Optimization over $L_2$ Ball

- (Deterministic) Optimization:

- Statistical Learning:

- Stoch. Aprx. / One-pass SGD:

- Online Learning (avg regret):
Questions

• What about other (convex) learning problems (other geometries):
  
  – Is Stochastic Approximation always optimal?
  
  – Are the rates for learning (# of samples) and optimization (runtime / # of accesses) always the same?
Outline

• **Deterministic Optimization vs Stat. Learning**
  – Main result: fat shattering as lower bound on optimization
  – Conclusion: sample complexity ≤ opt runtime

• **Stochastic Approximation for Learning**
  – Online Learning
  – Optimality of Online Mirror Descent

Very briefly
Optimization Complexity

\[ \min_{w \in \mathcal{W}} f(w) \]

- Optimization problem defined by:
  - Optimization space \( \mathcal{W} \)
  - Function class \( \mathcal{F} \subseteq \{ f: \mathcal{W} \rightarrow \mathbb{R} \} \)

- Runtime to get accuracy \( \epsilon \):
  - Input: instance \( f \in \mathcal{F}, \epsilon > 0 \)
  - Output: \( w \in \mathcal{W} \) s.t.
    \[ f(w) \leq \inf_{w \in \mathcal{W}} f(w) + \epsilon \]

- Count number of local black-box accesses to \( f(\cdot) \):
  \( O^f: \mathbb{R} \rightarrow f(w), \nabla f(w), \) any other “local” information
  \( (\forall \text{neighborhood } N(w) f_1 = f_2 \text{ on } N(w) \Rightarrow O^{f_1}(w) = O^{f_2}(w)) \)
Generalized Lipchitz Problems

$$\min_{w \in \mathcal{W}} f(w)$$

- **We will consider problems where:**
  - $\mathcal{W}$ is a convex subset of a vector space $\mathcal{L}$ (e.g. $\mathbb{R}^d$ or inf. dim.)
  - $\mathcal{X}$ convex $\subset \mathcal{L}^*$
  - $\mathcal{F} = \mathcal{F}_{\text{lip}(\mathcal{X})} = \{ f: \mathcal{W} \rightarrow \mathbb{R} \text{ convex} \mid \forall_w \nabla f(w) \in \mathcal{X} \}$

- **Examples:**
  - $\mathcal{X} = \{ |x|_2 \leq 1 \}$ corresponds to standard notion of Lipchitz functions
  - $\mathcal{X} = \{ |x| \leq 1 \}$ corresponds to Lipchitz w.r.t. norm $|x|$

- **Theorem (Main Result):**
  The $\epsilon$-fat shattering dimension of $\text{lin}(\mathcal{W},\mathcal{X})$ is a lower bound on the number of accesses required to optimize $\mathcal{F}_{\text{lip}}$ to within $\epsilon$
Fat Shattering

• Definition:

• $x_1, \ldots, x_m \in \mathcal{X}$ are $\epsilon$-fat shattered by $\mathcal{W}$ if there exists scalars $t_1, \ldots, t_n$ s.t. for every sign pattern $y_1, \ldots, y_m$, there exists $w \in \mathcal{W}$ s.t. $y_i(\langle w, x_i \rangle - t_i) > \epsilon$.

• The $\epsilon$-fat shattering dimension of $\text{lin}(\mathcal{W}, \mathcal{X})$ is the largest number of points $m$ that can be $\epsilon$-fat shattered
Optimization, ERM and Learning

• Supervised learning with linear predictors:
  \[ \hat{L}(w) = \frac{1}{m} \sum_{t=1}^{m} \text{loss}( \langle w, x_t \rangle, y_t ) \]

  1-Lipschitz
  \[ x_t \in \mathcal{X} \]

  ERM: \( \hat{w} = \min_{w \in \mathcal{W}} \hat{L}(w) \)

  Gradient of (empirical) risk: \( \nabla \hat{L}(w) \in \text{conv}(\mathcal{X}) \)

• Learning guarantee:
  If for some \( q \geq 2 \), \( \text{fat-dim}(\epsilon) \leq \frac{V}{\epsilon^q} \) \( \Rightarrow \)
  \( L(\hat{w}) \leq \inf_{w \in \mathcal{W}} L(w) + O( V \log^{1.5}(m) / m^{1/q} ) \)

• Conclusion:
  For \( q \geq 2 \), if there exists \( V \) s.t. the rate of optimization is at most
  \( \epsilon(m) \leq \frac{V}{T^{1/q}} \),
  then the statistical rate of the associated learning problem is at most:
  \( \epsilon(m) \leq 36 \sqrt[1.5]{V \log^{1.5}(m)} / m^{1/q} \)
Convex Learning ⇒ Linear Prediction

• Consider learning with a hypothesis class \( \mathcal{H} = \{ h : \mathcal{X} \rightarrow \mathbb{R} \} \)
  \( \hat{L}(h) = \frac{1}{m} \sum_{t=1}^{m} \text{loss}( h(x_t), y_t ) \)

• With any meaningful loss, \( \hat{L}(h_w) \) will be convex in a parameterization \( w \), only if \( h_w(x) \) is linear in \( w \), i.e.
  \( h_w(x) = \langle w, \phi(x) \rangle \)

• Rich variety of learning problems obtained with different (sometimes implicit) choices of linear hypothesis classes, feature mappings \( \phi \), and loss functions.
Linear Prediction

• Gradient space $\mathcal{X}$ is the learning data domain (i.e. the space learning inputs come from), or image of feature map $\phi$
  – $\phi$ specified via Kernel (as in SVMs, kernalized logistic or ridge regression)
  – In boosting: coordinates of $\phi$ are “weak learners”
  – $\phi$ can specify evaluations (as in collaborative filtering, total variation problems)

• Optimization space $F$ is the hypothesis class, the set of allowed linear predictors. Corresponds to choice of “regularization”
  – $L_2$ (SVMs, ridge regression)
  – $L_1$ (LASSO, Boosting)
  – Elastic net, other interpolations
  – Group norms
  – Matrix norms: trace-norm, max-norm, etc (eg for collaborative filtering and multi-task learning)

• Loss function need only be (scalar) Lipchitz.
  – hinge, logistic, etc
  – structured loss, where $y_i$ non-binary (CRFs, translation, etc)
  – exp-loss (Boosting), squared loss $\Rightarrow$ NOT globally Lipchitz
Main Result

• Problems of the form:
  \[ \min_{w \in \mathcal{W}} f(w) \]
  - \( \mathcal{W} \) convex \( \subset \) vector space \( \mathcal{B} \) (e.g. \( \mathbb{R}^n \), or inf.-dimensional)
  - \( \mathcal{X} \) convex \( \subset \mathcal{B}^* \)
  - \( f \in \mathcal{F} = \mathcal{F}_{\text{lip}(\mathcal{X})} = \{ f: \mathcal{W} \to \mathbb{R} \text{ convex} \mid \forall w \nabla f(w) \in \mathcal{X} \} \)

• Theorem (Main Result):
The \( \epsilon \)-fat shattering dimension of \( \text{lin}(\mathcal{W}, \mathcal{X}) \) is a lower bound on the number of accesses required to optimize \( f \in \mathcal{F}_{\text{lip}} \) to within \( \epsilon \)

• Conclusion:
  For \( q \geq 2 \), if for some \( V \), the rate of ERM optimization is at most
  \[ \epsilon(m) \leq V/T^{1/q}, \]
  then the learning rate of the associated problem is at most:
  \[ \epsilon(m) \leq 36 V \log^{1.5}(m) / m^{1/q} \]
Proof of Main Result

- **Theorem:**
  The $\epsilon$-fat shattering dimension of $\text{lin}(\mathcal{W},\mathcal{X})$ is a lower bound on the number of accesses required to optimize $\mathcal{F}_{\text{lip}}$ to within $\epsilon$.

- That is, for any optimization algorithm, there exists a function $f \in \mathcal{F}_{\text{lip}}$ s.t. after $m = \text{fat-dim}(\epsilon)$ local accesses, the algorithm is $\geq \epsilon$-suboptimal.

- **Proof overview:**
  View optimization as a game, where at each round $t$:
  - Optimizer asks for local information at $w^t$,
  - Adversary responds, ensuring consistency with some $f \in \mathcal{F}$.
  We will play the adversary, ensuring consistency with some $f \in \mathcal{F}$ where $\inf_w f(w) \leq \epsilon$, but where $f(w^t) \geq 0$. 

Playing the Adversary

• $x_1,..,x_m$ fat-shattered with thresholds $s_1,..,s_m$.
  I.e., $\forall$ signs $y_1,..,y_m$ $\exists$ w s.t. $y_i(\langle w, x_i \rangle - s_i) \geq \epsilon$

• We will consider functions of the form:
  $$f_y(w) = \max_i y_i(s_i - \langle w, x_i \rangle)$$

• Convex, piecewise linear
• (Sub)-gradients are $y_i x_i \Rightarrow f_y \in \mathcal{F}_{lip(\mathcal{X})}$
• Fat shattering $\Rightarrow \forall y \inf_w f_y(w) \leq -\epsilon$
Playing the Adversary

\[ f_y(w) = \max_i y_i(s_i - \langle w, x_i \rangle) \]

- **Goal**: ensure consistency with some \( f_y \) s.t. \( f_y(w^t) \geq 0 \)
- **How**: Maintain model
  \[ f^t(w) = \max_{i \in A^t} y_i(s_i - \langle w, x_i \rangle) \]
  based on \( A^t \subseteq \{1..m\} \).

  - Initialize \( A^0 = \{\} \)
  - At each round \( t=1..m \), add to \( A_t \):
    \[ i^t = \arg\max_{i \in A^{t-1}} |s_i - \langle w, x_i \rangle| \]
    and set corresponding \( y_i \) s.t. \( y_i(s_i - \langle w, x_i \rangle) \geq 0 \)
  - Return local information at \( w^t \) based on \( f^t \)

- **Claim**: \( f^t \) agrees with final \( f_y \) on \( w^t \), and so adversarial responses to algorithm are consistent with \( f_y \), but
  \[ f_y(w^t) = f^t(w^t) \geq 0 \geq \inf_w f_y(w) + \epsilon \]
Optimization vs Learning

\[ \text{(deterministic)} \]

\[
\text{Optimization} \geq d_\epsilon = \text{Statistical Learning} \]

\[ \text{runtime,} \]

\[ \text{# func, grad accesses} \]

- Converse?
  - Optimize with \(d_\epsilon\) accesses? (intractable alg OK)
  - Learning \(\Rightarrow\) Optimization?

With sample size \(m\), exact grad calculation is \(O(m)\) time, and so even if \#iter=\#samples, runtime is \(O(m^2)\).

- Stochastic Approximation?
  (stochastic, local access, \(O(1)\) memory method)
Online Optimization / Learning

• Online optimization setup:
  – As before, problem specified by $\mathcal{W}, \mathcal{F}$
  – $f_1, f_2, \ldots$ presented sequentially by “adversary”
  – “Learner” responds with $w_1, w_2, \ldots$
    
    Adversary:
    
    Learner:
    
  – Formally, learning rule $A: \mathcal{F}^* \to \mathcal{W}$ with $w_t = A(f_1, \ldots, f_{t-1})$

• Goal: minimize regret versus best single response in hindsight.
  – Rule $A$ has regret $\epsilon(m)$ if for all sequences $f_1, \ldots, f_m$:
    \[
    \frac{1}{m} \sum_{t=1}^{m} f_t(w_t) \leq \inf_{w \in \mathcal{W}} \frac{1}{m} \sum_{t=1}^{m} f_t(w) + \epsilon(m)
    \]

• Examples:
  – Spam Filtering
  – Investment return:
    
    $w[i]$ = investment in holding $i$
    
    $f_t(w) = -\langle w, z_t \rangle$, where $z_t[i]$ = return on holding $i$
Online To Batch

- An online optimization algorithm with regret guarantee
\[
\frac{1}{m} \sum_{t=1}^{m} f_t(w_t) \leq \inf_{w \in W} \frac{1}{m} \sum_{t=1}^{m} f_t(w) + \epsilon(m)
\]
can be converted to a learning (stochastic optimization) algorithm, by running it on a sample and outputting the average of the iterates: [Cesa-Bianchi et al 04]:
\[
\mathbb{E} [L(\bar{w}_m)] \leq \inf_{w \in W} L(w) + \epsilon(m)
\]
\[
\bar{w}_m = \frac{w_1 + \ldots + w_m}{m}
\]
(in fact, even with high probability rather then in expectation)

- An online optimization algorithm *that uses only local info* at \(w_i\) can also be used as for deterministic optimization, by setting \(z_i=z\):
\[
f(\bar{w}_m) \leq \inf_{w \in W} f(w) + \epsilon(m)
\]
Online Gradient Descent

\[ w_{t+1} \leftarrow \Pi_{W} (w_t - \eta_t \nabla_w f(w_t, z_t)) \]

- **Regret guarantee:**
  \[
  \frac{1}{m} \sum_{t=1}^{m} f_t(w_t) \leq \frac{1}{m} \sum_{t=1}^{m} f_t(w^*) + \sqrt{\frac{R^2 B^2}{m}}
  \]
  where
  - \( B = \sup_{w \in W} ||w||_2 \)
  - \( R = \sup_{w \in W, f \in F} ||\nabla_w f(w)||_2 \)

- **Online To Stochastic Conversion** ⇒ Stochastic Gradient Descent
- **Online to Deterministic Conversion** ⇒ Gradient Descent

---

Onlined Gradient Descent

[Zinkevich 03]

online2stochastic

[Cesa-Binachi et al 04]

Stochastic Gradient Descent

[Nemirovski Yudin 78]
Classes of Optimization/Learning Problems

- Problem specified by:
  - Optimization space / Hypothesis class $\mathcal{W}$
  - Function class $\mathcal{F} = \{ f: \mathcal{W} \to \mathbb{R} \}$

- For convex $\mathcal{W} \subset B$ and $\mathcal{X} \subset B^*$, we consider:

  $\mathcal{F}_{\text{lip}} = \{ f(w) \mid \forall_w \nabla f(w) \in \mathcal{X} \}$

  $\mathcal{F}_{\text{sup-abs}} = \{ f_{x,y}(w) = |\langle w, x \rangle - y| \mid x \in \mathcal{X}, y \in \mathbb{R} \}$
  or $\mathcal{F}_{\text{sup-hinge}} = \{ f_{x,y}(w) = [1 - y \langle w, x \rangle]_+ \mid x \in \mathcal{X}, y = \pm 1 \}$

  $\mathcal{F}_{\text{lin}} = \{ f_x(w) = \langle w, x \rangle \mid x \in \mathcal{X} \}$

- For all the above, $\mathcal{X}$ specifies the possible subgradients $\nabla f(w)$

  $\mathcal{F}_{\text{lin}}, \mathcal{F}_{\text{sup}} \subseteq \mathcal{F}_{\text{lip}}$
Optimization vs Learning

Online Optimization (of $\mathcal{F}_{\text{lip}}$) with Local Info ≥ Deterministic, Local-Access Optimization (of $\mathcal{F}_{\text{lip}}$) ≥ Stat Learning (Stoch Opt of $\mathcal{F}_{\text{sup}}$)

runtime,
# func, grad accesses
# samples, full access

• For $L_2$ geometry ($\mathcal{X}=\{||x||_2 \leq R\}$, $\mathcal{W}=\{||x||_2 \leq B\}$): Online/Stoch Grad Descent
  – Optimal for Learning
  – local access ($1^{st}$ order), $O(1)$ memory, optimizes $\mathcal{F}_{\text{lip}}$
Online Mirror Descent

• Grad Descent is inherently related to $L_2$ norm.
• To handle other geometries (other $\mathcal{W}$, $\mathcal{X}$), consider potential function (regularizer) $\Psi: \mathcal{W} \rightarrow \mathbb{R}$ and the Bergman Divergence:

$$D_\Psi(w, v) = \Psi(w) - \Psi(v) - \langle \nabla \Psi(v), w - v \rangle$$

• We will need $\Psi$ that is non-negative and $q$-uniformly convex w.r.t. $|| \cdot ||_{\mathcal{X}^*}$ on $\mathcal{W}$, i.e. s.t. for all $v, w \in \mathcal{W}$:

$$D_\Psi(w, v) \geq 1/q (||w - v||_{\mathcal{X}^*})^q$$

• Online Mirror Descent:

$$w_{t+1} \leftarrow \text{arg min}_{w \in \mathcal{W}} \eta_t \langle \nabla f_t(w_t), w \rangle + D_\Psi(w, w_t)$$

• Regret Guarantee:

$$\frac{1}{m} \sum_{t=1}^m f_t(w_t) \leq \frac{1}{m} \sum_{t=1}^m f_t(w^*) + 2 \sqrt{\frac{\sup_{w \in \mathcal{W}} \Psi(w)}{m}}$$

as long as $\nabla f(w) \in \mathcal{X}$

[Nemirovski Yudin 78] [Beck Teboulle 03] [S Sridharan Tewari 11]
Optimality of Online Mirror Descent

• Theorem:
  For any convex centrally symmetric $\mathcal{X}$, $\mathcal{W}$, if there exists an online learning rule for $\mathcal{F}_{\text{sup}}$ (or $\mathcal{F}_{\text{lin}}$ or $\mathcal{F}_{\text{lip}}$) with online regret
  $$\epsilon(m) \leq \frac{V}{m^{1/q}}$$
  then there exists $\Psi$ and step size $\eta$, s.t. the regret of online Mirror Descent on $\mathcal{F}_{\text{lip}}$ (and so also $\mathcal{F}_{\text{sup}}$, $\mathcal{F}_{\text{lin}}$) is at most:
  $$\epsilon_{\text{MD}}(m) \leq 6002 \log^2(m) \frac{V}{m^{1/q}}$$

[S Sridharan Tewari 11]
Optimization vs Learning

Mirror Descent

Deterministic, Local-Access Optimization (of $F_{\text{lip}}$)

Online Learning ($\mathcal{F}_{\text{sup}}$)

# samples

Stat Learning (Stoch Opt of $\mathcal{F}_{\text{sup}}$)

Online Optimization (of $\mathcal{F}_{\text{lip}}$)

# samples

- Mirror Descent is (nearly) optimal whenever online learning is possible (i.e. ensuring small adversarial regret).

- For such problems, need only consider Online/Stochastic Mirror Descent, a local (1st order), $O(1)$ memory, SA-type method.
Summary

Tight connections between learning and optimization:

- Learning IS Optimization
- Fat shattering as lower bound on deterministic optimization runtime
- Mirror Descent optimal for Online Learning