The Sample Complexity in Data(Sample)-Driven Optimization

Yinyu Ye

Joint work with many others

Department of Management Science and Engineering
and ICME
Stanford University
Outline

1. Sample Complexity in Reinforcement Learning and Stochastic Game
   - Reinforcement Learning and MDP
   - The Sample-Complexity Problem in RL
   - The Near Sample-Optimal Algorithm
     - Variance Reduction
     - Monotonicity Analysis

2. Average Approximation with Sparsity-Inducing Penalty for High-Dimensional Stochastic Programming/Learning
   - Regularized sample average approximation (RSAA)
   - Theoretical generalizations
   - Theoretical applications:
     - High-dimensional statistical learning
     - Deep neural network learning
I. What Is Reinforcement Learning?

• Use samples/experiences to control an unknown system

Samples/Experiences: previous observations
Reinforcement learning achieves phenomenal empirical successes

Hessel et al’17

Median over 57 Atari games

Human

Median human-normalized score

Millions of frames
What if sample/trial is costly and limited?

Sample Size/Complexity is essential!
“What If” Model: Markov Decision Process

- States: $S$  Actions: $A$
- Reward: $r(s, a) \in [0, 1]$
- State transition: $P(s'|s, a)$

- Policy: $\pi : S \rightarrow A$

$$\max_{\pi} v^\pi := \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t r(s^t, a^t) \right]$$

$\gamma \rightarrow 1$  Effective Horizon: $(1 - \gamma)^{-1}$

- Optimal policy & value: $\pi^* \quad V^*$
- $\epsilon$-optimal policy $\pi$: $V^* - V^{\pi} \leq \epsilon$

When model is known, solve via Linear Programming
Examples

• Three-States MDP
  - Optimal policy: \( \pi^*(1) = L \)
  - 0.1-optimal policy: \( \pi(1) = R \)

• Pacman
  - How to get an 0.1-optimal policy?
  - Learn the policy from sample data
    - Short episodes:
      \( (s_1, a_1, s'_1), (s_2, a_2, s'_2), (s_3, a_3, s'_3), \ldots \)
For an $M = (S, A, P, r, \gamma)$ how many samples are **sufficient/necessary** to **learn** a **0.1-optimal** policy w. p. $\geq 0.9$?

Data/sample: transition examples from every $(s,a)$

$$(s_1, a_1, s'_1), (s_2, a_2, s'_2), (s_3, a_3, s'_3), \ldots$$

• $(1 - \gamma)^{-3}$ samples per $(s,a)$ is necessary!
• No previous algorithms achieves this sample complexity
• Sufficient side: what is the **best** way of learning policy?

[Azar et al (2013)]
Empirical Risk Minimization and Value-Iteration

- Collect $m$ samples for each $(s, a)$
- Construct an empirical model
- Solve the empirical MDP via Linear Programming or Simply Value-Iteration:

$$v_s \leftarrow \max_a \left[ r(s,a) + \hat{P}(s'|s,a) v_{s'} \right] \text{ for all } s$$

**Sample-First and Solve-Second**

- Simple and intuitive
- Space: model-based, all data
- Samples: $m \geq (1 - \gamma)^{-5}$

$$\gamma = 0.95, \quad m \geq 3,000,000$$

[Azar et al’ 2013]
Iterative Q-learning: Solving while Sampling

\[ Q(s, a) : \text{quality of (s,a), initialized to } 0 \]

\[ Q^* : \pi^*(s) = \arg\max_a Q^*(s, a) \]

\[ Q(s, a) \leftarrow r(s, a) + \gamma \hat{E}_1[Q|s, a] \]

Converging to \( Q^* \)

Empirical estimator of the expected future value

\[ \hat{E}_1[Q|s, a] = \frac{1}{m_1} \sum_{i=1}^{m_1} \max_{a'} Q(x^{(i)}, a') \]

\[ x^{(i)} \sim P(\cdot|s, a) \]

After \((1 - \gamma)^{-1}\) iterations:

\[ \pi(s) \leftarrow \arg\max_a Q(s, a) \]

• #Samples for 0.1-opt. policy w.h.p.:

\[ m \geq (1 - \gamma)^{-7} \]

\[ \gamma = 0.95, m \geq 1,000,000,000 \]
Why Iterative Q-Learning Fails

\[ Q(s, a) \leftarrow r(s, a) + \gamma \hat{E}_1[Q|s, a] \]

“a little closer” to \( Q^* \)
\[ \|Q^{i+1} - Q^*\|_\infty \leq \gamma \|Q^i - Q^*\|_\infty \]

Empirical estimator of the expected future value

- Runs for \( R = (1 - \gamma)^{-1} \) iterations
- Each iteration uses new samples of same number

Too many samples for too little improvement
Summary of Previous Methods

How many samples per \((s, a)\) are **sufficient/necessary** to obtain a 0.1-**optimal** policy w. p. > 0.9?

<table>
<thead>
<tr>
<th>Previous Algorithms</th>
<th>#Samples/(s,a)</th>
<th>(\gamma = 0.99)</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phased Q-learning</td>
<td>((1 - \gamma)^{-7})</td>
<td>10^{14}</td>
<td>Kearns &amp; Singh’ 1999</td>
</tr>
<tr>
<td>ERM</td>
<td>((1 - \gamma)^{-5})</td>
<td>10^{10}</td>
<td>Azar et al’ 2013</td>
</tr>
<tr>
<td>Randomized LP</td>
<td>((1 - \gamma)^{-4})</td>
<td>10^{8}</td>
<td>Wang’ 2017</td>
</tr>
<tr>
<td>Randomized Val. Iter.</td>
<td>((1 - \gamma)^{-4})</td>
<td>10^{8}</td>
<td>Sidford et al’ ISML2018</td>
</tr>
</tbody>
</table>

Previous Best Algorithm: \((1 - \gamma)^{-4}\) **Necessary**?

Information theoretical lower bound: \(\Omega[(1 - \gamma)^{-3}]\) **Sufficient**? \(\gamma \rightarrow 1\)
The Near Sample-Optimal Algorithm

**Theorem (informal):**

For an unknown discounted MDP, \( M = (S, A, P, r, \gamma) \) model size \(|S|^2|A|\).

We give an algorithm (vQVI) that

- Recover w.h.p. an 0.1-optimal policy with \( \tilde{\Theta}[(1 - \gamma)^{-3}] \) samples per \((s,a)\)
- Time: \#samples
- Space: model-free

The first sample-optimal* algorithm for learning policy of DMDP.

[Sidford, Wang, Wu, Yang, and Ye, NeurIPS (2018)]
Analysis Overview

Q-Learning

\[ Q(s, a) \leftarrow r(s, a) + \gamma \hat{E}_1[Q|s, a] \]

- **Variance Reduction**
- **Monotonicity**
- **Bernstein**

### Utilizing previous estimate
- **Q-Learning**
  \[ (1 - \gamma)^{-7} \]
- **Var. Reduced Q-Learning**
  \[ (1 - \gamma)^{-6} \]

### Guarantee good policy
- **Monotonicity**
  \[ (1 - \gamma)^{-4} \]

### Precise control of error
- **Bernstein + Law of total var.**
  \[ (1 - \gamma)^{-3} \]
Variance Reduction

• Original Q-Learning

\[ Q(s, a) \leftarrow r(s, a) + \gamma \hat{E}_1[Q|s, a] \]

“a little closer” to \( Q^* \)

• Use milestone-point \( Q^0 \) to reduce samples used

\[ Q(s, a) \leftarrow r(s, a) + \gamma \hat{E}_1[Q - Q^0|s, a] + \gamma \hat{E}_0[Q^0|s, a] \]

Compute many times; each time uses a small number of samples
Compute once; reused for many iterations

\[ \hat{E}_1[Q - Q^0|s, a] = \frac{1}{m_1} \sum \text{[random variables with small variance]} \]
Algorithm: vQVI (sketch)

\[ Q(s, a) \leftarrow r(s, a) + \gamma \hat{E}_1 [Q - Q^0 | s, a] + \gamma \hat{E}_0 [Q^0 | s, a] \]

Check-point

\[ w^0(s, a) \approx \hat{E}_0(Q^0 | s, a) = \frac{1}{m_0} \sum_{k=1}^{m_0} \max_{a' \in A} Q^i(x^{(k)}, a') \]

Large number of samples

\[ Q^{i+j}(s, a) \approx r(s, a) + \gamma \hat{E}[Q^{i+j-1} - Q^i | s, a] + \gamma w^0(s, a) \]

Small number of samples

\[ H = (1 - \gamma)^{-1} \text{ iters} \]

\[ \| Q^{i+H} - Q^* \|_\infty \leq \| Q^i - Q^* \|_\infty / 2 \]

"Big improvement"
Compare to Q-Learning

- Q-learning variants

<table>
<thead>
<tr>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
</tr>
</thead>
</table>

#Samples: (minibatch size) \times R

\[ R = (1 - \gamma)^{-1} \]

- Variance-reduced Q-Learning

#samples: \approx \text{single minibatch}
Precise Control of Error Accumulation

• Each modified Q-learning step has estimation error

\[ Q^{i+j}(s, a) \approx r(s, a) + \gamma \hat{E}[Q^{i+j-1} - Q^i | s, a] + \gamma w^0(s, a) \]

• Error

\[ \epsilon^{i+j} = |Q^{i+j} - E(Q^{i+j})| \]

• Bernstein inequality + Law of total variance

\[ \sum_{j=1}^{H} \epsilon^{i+j} \sim \sqrt{\frac{\text{total variance of MDP}}{m}} \leq \sqrt{(1 - \gamma)^{-3}/m} \]

the intrinsic complexity of MDP
Extension to Two-Person Stochastic Game

- Fundamentally different computation classes even for the discount case.
  
  [Sidford, Wang, Yang, Ye (2019)]: \( \tilde{\Theta}[(1 - \gamma)^{-3}] \) samples per \((s, a)\) to learn 0.1-opt strategy w.h.p.

- But they achieve the same complexity lower bound for the discount case!
II. Stochastic Programming

• Consider a Stochastic Programming (SP) problem

\[
\min_{x \in \mathbb{R}^p : 0 \leq x \leq R} F(x) = \mathbb{E}_p f(x, W)
\]

• \(f\) deterministic. For every feasible \(x\), the function \(f(x, \cdot)\) is measurable and \(\mathbb{E}_p f(x, W)\) is finite. \(W\) is the support of \(W\).

• Commonly solved Sample Average Approximation (SAA), a.k.a., Monte Carlo (Sampling) method.
 Sampling Average Approximation

• To solve

\[ x^{true} \in \text{argmin}\{F(x): 0 \leq x \leq R\} \]

• Solve instead an approximation problem

\[ \min_{x \in \mathbb{R}^p: 0 \leq x \leq R} \left\{ \hat{F}_n(x) := n^{-1} \sum_{i=1}^{n} f(x, W_i) \right\} \]

• \{W_1, W_2, ..., W_i, ... W_n\} is a sequence of samples of W
• Simple to implement
• Often tractably computable

SAA is a popular way of solving SP
### Equivalence between SAA and M-Estimation

<table>
<thead>
<tr>
<th>Stochastic Programming</th>
<th>Statistical learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAA</td>
<td>Model Fitting Formulation for M-Estimation</td>
</tr>
<tr>
<td>RSAA</td>
<td>Regularized M-Estimation</td>
</tr>
<tr>
<td>Suboptimality gap $\epsilon$</td>
<td>Excess Risk: $F(\cdot) - F(x^{true})$</td>
</tr>
<tr>
<td>Minimizer $x^{true}$</td>
<td>True parameters $x^{true}$</td>
</tr>
<tr>
<td>Function</td>
<td>Statistical Loss</td>
</tr>
</tbody>
</table>

Bach & Moulines, 2011. *NeuIPS*
Efficacy of SAA

- **Assumptions:**
  - (a) The sequence \( \{f(x, W_1), f(x, W_2), \ldots, f(x, W_n)\} \) i.i.d.
  - (b) Subgaussian \((\text{can be generalized into sub-exponential})\)
  - (c) Lipschitz-like condition

- Number \( n \) of samples required to achieve \( \epsilon \) accuracy with probability \( 1 - \alpha \) in solving an \( p \)-dimensional problem.
  \[
  \Pr[\mathbf{F}(\mathbf{x}^{SAA}) - \mathbf{F}(\mathbf{x}^{true}) \leq \epsilon] \geq 1 - \alpha
  \]

- If \( n \) is large enough to satisfy
  \[
  n \geq \frac{p}{\epsilon^2} \ln \frac{1}{\epsilon} + \frac{1}{\epsilon^2} \ln \frac{1}{\alpha}
  \]

Sample size \( n \) grows polynomially when number of dimensions \( p \) increases

Shapiro et al., 2009, Shapiro, 2003, Shapiro and Xu, 2008
High-Dimensional SP

• What if $p$ is large, but acquiring new samples are prohibitively costly?

• What if most dimensions are (approximately) ineffective, e.g., $x^{true} = [80; 131; 100; 0; 0; \ldots; 0]$

• Consider a problem with $p = 100,000$
  • SAA requires $n \geq 100,000 \sim 10,000,000$
  • If only it is known which dimensions (e.g., three of them) are nonzero, then original problem can be reduced to $p = 3$, then $n \geq 10 \sim 1,000$

Conventional SAA is not effective in solving high-dimensional SP
Convex and Nonconvex Regularization Schemes

RSAA formulation: \( \min_{x: \ 0 \leq x \leq R} F_{n, \lambda}(x) := \hat{F}_n(x) + \sum_{j=1}^{p} P_{\lambda}(x_j) \)

- \( l_q \)-norm (0 < \( q \leq 1 \) or \( q = 2 \)), biased or non-sparse
  - \( P_{\lambda}(t) := \lambda \cdot t^q \)

- \( l_0 \)-norm (\( \cdot \)), not continuous
  - \( P_{\lambda}(t) := \lambda \cdot \mathbb{1}(t \neq 0) \)

- Folded concave penalty in the form of minimax concave penalty (\( \cdot \))
  - \( P_{\lambda}(t) := \int_{0}^{t} \frac{(a\lambda - s)^+}{a} \, ds \)

FCP entails unbiasedness, sparsity, and continuity

Frank and Friedman, 1993. *Technometrics*
Fan and Li, 2001. *JASA*
Raskutti et al., 2011, *IEEE Trans. Inf. Theory*
Existing Results on Convex Regularization

- **Lasso** (Tibshirani, 1996), *compressive sensing* (Donoho, 2006), and *Dantzig selector* (Candes & Tao, 2007):

  - **Pro:**
    - Computationally tractable
    - Generalization error and oracle inequalities (e.g., \(\ell_1, \ell_2\)-loss): Bunea et al. (2007), Zou (2006), Bickel et al. (2009), Van de Geer (2008), Negahban et al. (2010), etc.

  - **Cons:**
    - Requirement of RIP, restricted eigenvalue (RE), or RSC
    - Absence of (strong) oracle property: Fan and Li (2001)
    - Biased: Fan and Li (2001)

  - **Remark**
    - Excess risk bound available for linear regression beyond RIP, RE, or RSC (e.g., Zhang et al., 2017)
Existing Results on (Folded) Concave Penalty


- Pro:
  - Generalization error ($\ell_1, \ell_2$-loss) and oracle inequalities: Loh and Wainwright (2015), Wang et al. (2014), Zhang & Zhang (2012), etc.

- Cons:
  - Strong NP-Hard, requirement of RSC, except for the special case of linear regression

- Remark:
  - Chen, Fu and Y (2010): threshold lower-bound property at every local solution;
  - Zhang and Zhang (2012): global optimality results in good statistical quality;
  - Fan and Lv (2011): sufficient condition results in desirable statistical property;
The $S^3$ONC Solutions Admits FPTAS

**Assumption (d):**

\[ f(\cdot, z) \text{ is continuously differentiable and for a.e. } z \in \mathcal{W}, \]

\[
\left| \left[ \frac{\partial \hat{F}_n(x, z)}{\partial x_j} \right]_{x=\bar{x}+\delta e_j} - \left[ \frac{\partial \hat{F}_n(x, z)}{\partial x_j} \right]_{x=\bar{x}} \right| \leq U_L \cdot |\delta|
\]

- First-order necessary condition (FONC): The solution $x^* \in [0, R]^p$

  \[
  \langle \nabla \hat{F}_n(x^*) + (P'_{\lambda}(x_j^*): 1 \leq j \leq p), x - x^* \rangle \geq 0, \quad \forall x \in [0, R]^p
  \]

- Significant subspace second-order necessary condition ($S^3$ONC):
  - FONC holds, and, for all $j = 1, \ldots, p$: $x_j^* \in (0, a\lambda)$, it holds that

    \[
    U_L + P''_{\lambda}(x_j^*) \geq 0 \quad \text{or} \quad \nabla^2 \hat{F}_{jj}(x^*) + P''_{\lambda}(x_j^*) \geq 0
    \]

- $S^3$ONC is weaker than the canonical second-order KKT
  - Computable within pseudo-polynomial time

Result 1: Efficacy of RSAA at a Global Minimum

THEOREM 1

Assumptions:
- (a) – (d)

Consider a global minimal solution $x^*$ to RSAA. If the sample size $n$ satisfies

$$n \geq \frac{s^{2.5}}{\varepsilon^3} \left( \ln \frac{p}{\varepsilon} \right)^{1.5} + \frac{1}{\varepsilon^2} \ln \frac{1}{\alpha},$$

then $\mathbb{P}[F(x^*) - F(x^{true}) \leq \varepsilon] \geq 1 - \alpha$, for any $\alpha > 0$ and $\varepsilon \in (0, 0.5]$.

- Better dependence in $p$ than SAA: $\frac{p}{\varepsilon^2} \ln \frac{1}{\varepsilon} + \frac{1}{\varepsilon^2} \ln \frac{1}{\alpha}$
- However, generating a global solution is generally computationally prohibitive

Liu, Wang, Yao, Li, Ye, 2018. *Math Program*
Result 2: Efficacy of RSAA at a Local Solution

THEOREM 2

Assumptions:
• (a) – (d)
• $f(\cdot, z)$ is convex for almost every $z \in \mathcal{W}$

Consider an $S^3$ONC solution $x^*$ to RSAA. If $F_{n,\lambda}(x^*) \leq F_{n,\lambda}(0)$ a.s., and if the sample size $n$ satisfies

$$n \gtrsim \frac{S^{2.5}}{\epsilon^4} \left( \ln \frac{p}{\epsilon} \right)^2 + \frac{1}{\epsilon^2} \ln \frac{1}{\alpha},$$

then $\Pr[F(x^*) - F(x^{true}) \leq \epsilon] \geq 1 - \alpha$, for any $\alpha > 0$ and $\epsilon \in (0, 0.5]$.

• Better dependence in $p$ than SAA: $\frac{p}{\epsilon^2} \ln \frac{1}{\epsilon} + \frac{1}{\epsilon^2} \ln \frac{1}{\alpha}$
• Much easier solvable than a global minimizer: Pseudo-polynomial-time algorithms exist

Result 3: Efficacy of RSAA Under Additional Assumptions

THEOREM 3

Assumptions:
• (a) – (d)
• $f(\cdot, z)$ is convex for almost every $z \in \mathcal{W}$
• $\mathbf{F}$ is strongly convex and differentiable (*)

Consider an $S^3$ONC solution $\mathbf{x}^*$. If $F_{n,\lambda}(\mathbf{x}^*) \leq F_{n,\lambda}(0)$ a.s., and if the sample size $n$ satisfies

$$n \geq \frac{s^{1.5}}{\epsilon^3} \left( \frac{\ln p}{\epsilon} \right)^{1.5} + \frac{1}{\epsilon^2} \ln \frac{1}{\alpha},$$

then $\mathbb{P}[\mathbf{F}(\mathbf{x}^*) - \mathbf{F}(\mathbf{x}^{true}) \leq \epsilon] \geq 1 - \alpha$, for any $\alpha > 0$ and $\epsilon \in (0, 0.5]$.

• For some problems, while $f(\cdot, \mathcal{W})$ is only convex, $\mathbf{F}$ can be strongly convex, e.g., high-dimensional linear regression with random design matrix
• Under additional assumption (*), Theorem 3 shows an improved sample complexity than Theorem 2

Liu, Wang, Yao, Li, Ye, 2018. Math Program
Result 4: Efficacy of RSAA Under Additional Assumptions

**THEOREM 4**

Assumptions:
- (a) – (d)
- $f(\cdot, z)$ is convex for almost every $z \in \mathcal{W}$
- $F$ is strongly convex and differentiable (*)
- $\min\{|x_j^{true}| : x_j^{true} \neq 0, j = 1, \ldots, p\} \geq \text{Threshold} (**)$

Consider an $S^3$ONC solution $x^*$. If $F_{n,\lambda}(x^*) \leq F_{n,\lambda}(0)$ a.s., and if the sample size $n$ satisfies

$$n \geq \frac{s}{\epsilon^2} \ln \frac{p}{\epsilon} + \frac{1}{\epsilon^2} \ln \frac{1}{\alpha},$$

then $\mathbb{P}[F(x^*) - F(x^{true}) \leq \epsilon] \geq 1 - \alpha$, for any $\alpha > 0$ and $\epsilon \in (0, 0.5]$.

Under additional assumptions (*) and (**), no compromise on the rate in $\epsilon$ compared to the SAA: $\frac{p}{\epsilon^2} \ln \frac{1}{\epsilon} + \frac{1}{\epsilon^2} \ln \frac{1}{\alpha}$. 

Liu, Wang, Yao, Li, Ye, 2018. *Math Program*
Numerical Experiments

Fix sample size $n = 100$; Increase dimensionality

Fix dimension $p = 100$; Increase sample size

Liu, Wang, Yao, Li, Ye, 2018. Math Program
Generalizations and Applications

- **Theoretical Generalizations**
  - From subgaussian distribution to subexponentiality
  - From smooth to nonsmooth cost function

- **Theoretical Applications**
  - High-Dimensional Statistical Learning (HDSL)
  - Deep Neural Networks
Theoretical Generalizations

• Generalization from subgaussian distribution to subexponentiality

**Assumption (e):**

\[ f(x, W) \text{ is subexponential for all } x \in [0, R]^p \]

• Generalization from smooth cost function \( f \) to non-smooth cost function

**Assumption (f):**

\[ f(x, W) = f_1(x, W) + \max_{u \in \mathcal{H}} \{u^T A(W)x - \hat{\phi}(u, W)\} \]

where \( A(W) \in \mathbb{R}^{m \times p} \) is a linear operator, \( \hat{\phi} \) is a measurable, deterministic function, \( f_1(\cdot, W) \) is continuously differentiable with Lipschitz continuous gradient, \( \mathcal{H} \subset \mathbb{R}^m \) is a convex and compact set.
Result 5: Generalization to subexponentiality

Let $x^{\ell_1} := \arg \min_{x: 0 \leq x \leq R} \hat{F}_n(x) + \lambda \sum_{j=1}^{p} |x_j|$. 

THEOREM 5

Assumptions: (a), (c), (d), (e) subexponentiality; $f_1(\cdot, z)$ is convex for almost every $z \in \mathcal{W}$

Consider an $S^3$ONC solution $x^*$. If $F_{n,\lambda}(x^*) \leq F_{n,\lambda}(x^{\ell_1})$ a.s., and if the sample size $n$ satisfies

$$n \gtrsim \frac{s^3}{\epsilon^3} \left( \frac{\ln \frac{p}{\epsilon}}{\epsilon} \right)^{1.5} + \ln \frac{p}{\alpha} + \left( \frac{\ln 1}{\alpha} \right)^3,$$

then $\mathbb{P}[F(x^*) - F(x^{true}) \leq \epsilon] \geq 1 - \alpha$, for any $\alpha > 0$ and $\epsilon \in (0, 0.5]$.

- **Better dependence in $\epsilon$** than Theorem 2: $\frac{s^{2.5}}{\epsilon^4} \left( \frac{\ln \frac{p}{\epsilon}}{\epsilon} \right)^2 + \frac{1}{\epsilon^2} \ln \frac{1}{\alpha}$

- Desired solution can be generated by solving for an $S^3$ONC solution initialized with $x^{\ell_1}$.

Result 6: Generalization to non-smoothness

Let \( x^{\ell_1} := \arg \min_{x:0 \leq x \leq R} \hat{F}_n(x) + \lambda \sum_{j=1}^{p} |x_j| \).

THEOREM 6

Assumptions: (a), (c), (e) subexponentiality; (f); \( f(\cdot, z) \) is convex for almost every \( z \in \mathcal{W} \).
Consider an S\(^3\)ONC solution \( x^* \). If \( F_{n,\lambda}(x^*) \leq F_{n,\lambda}(x^{\ell_1}) \) a.s., and if the sample size \( n \) satisfies

\[
n \gtrsim \frac{s^4}{\epsilon^4} \left( \ln \frac{p}{\epsilon} \right)^2 + \ln \frac{p}{\alpha} + \left( \ln \frac{1}{\alpha} \right)^2,
\]

then \( \mathbb{P}[F(x^*) - F(x^{min}) \leq \epsilon] \geq 1 - \alpha \), for any \( \alpha > 0 \) and \( \epsilon \in (0, 0.5] \).

- Assumption (d) is replaced by (f)
- Desired solution can be generated by solving for an S\(^3\)ONC solution initialized with \( x^{\ell_1} \)
Theoretical Application 1: High-Dimensional M-Estimation

- Given knowledge of a statistical loss function \( f \), and hypothetically \( x^{true} \in \arg \min \{ F(x) : = \mathbb{E}[f(x,W)] \} \)

  \( x^{true} \): True parameters governing data generation model

- Estimate \( x^{true} \in \mathbb{R}^p \) given only \( f \) samples \( W_1, W_2, ..., W_n \).

- Assume that \( f(x,W_i), i = 1, ..., n \), is i.i.d. subexponential.

- Traditional approach: minimizing empirical risk to obtain unbiased estimator

  \[ \hat{x} = \arg \min \left[ \hat{F}_n(x) : = n^{-1} \sum_{i=1}^{n} f(x,W_i) \right] \]

- Under High-dimensional setting, \( p \gg n \)

- Consider Folded concave penalized (FCP) learning

  \[ \min \left\{ F_{n,\lambda}(x) : = n^{-1} \sum_{i=1}^{n} f(x,W_i) + \sum_{j=1}^{p} P_\lambda(|x_j|) \right\} \]
Result 7: Learning Under Approximate Sparsity

- **Approximate sparsity:** Existence of \( \tilde{x} \) such that
  \[
  F(\tilde{x}) - F(x^{true}) \leq \hat{\epsilon}, \quad \text{and} \quad s := ||\tilde{x}||_0 \ll p
  \]

**THEOREM 7**

For any \( \Gamma \geq 0 \), consider an S\(^3\)ONC solution \( x^* \) with \( F_{n,\lambda}(x^*) \leq \min_x F_{n,\lambda}(x) + \Gamma \), a.s. Assume

(a) subexponentiality; (b) Lipschitz-like continuity of \( F \) and \( \nabla F \); (c) approximate sparsity

\[
\text{Excess risk of } x^* \leq \tilde{O}(1) \cdot \left( \frac{s \sqrt{\ln p}}{n^{1/3}} + \frac{\Gamma + \hat{\epsilon}}{n^{1/3}} + \Gamma + \hat{\epsilon} \right)
\]

with probability \( 1 - O(1) \cdot \exp(-O(1)n^{1/3} \ln p) \)

- **First explication:** more general assumptions than sparsity;
  - \( \Gamma \) can be interpreted as the sub-optimality gap of \( x^* \)
  - No convexity or RSC assumed.

Removing $\Gamma$ via Wise Initialization for Convex Loss

**Step 1:** Solve the LASSO program

$$x^{\ell_1} := \arg\min_x \hat{F}_n(x) + \lambda \sum_{j=1}^p |x_i|$$

**Step 2:** Solve for an $S^3$ONC stationary point initialized with $\beta^{\ell_1}$

- A modified first-order method, ADMM

Algorithm-independent, fully polynomial-time approximation scheme

*Complexity $\leq \text{Poly}(p, \text{desired\_accuracy})$*
Consider an $S^3$ONC solution $x^*$ with $F_{n,\lambda}(x^*) \leq F_{n,\lambda}(x^\ell_1)$, a.s. Assume (a) subexponentiality; (b) Lipschitz-like continuity of $F$ and $\nabla F$; (c) sparsity (i.e., $\hat{\varepsilon} = 0$ in the assumption of approximate sparsity)

$$\text{Excess risk of } x^* \leq \tilde{O}(1) \cdot \left(\frac{s\|x^{true}\|_\infty \sqrt{\ln p}}{n^{1/3}} + \frac{s \ln p}{n^{2/3}}\right)$$

with probability $1 - O(1) \cdot \exp(-O(1)n^{1/3} \ln p)$

**THEOREM 8**

A pseudo-polynomial-time computable theory without RSC

Comparing This Research with Existing Results

- New insights on local solutions for folded concave penalty
- **Pros:**
  - No requirement of RSC, RIP, RE, or alike
  - Applicable to non-smooth learning and deep learning
  - Applicable to high-dimensional stochastic programming
  - Generalization error bounds on excess risk
- **Cons:**
  - Slower rate than alternative results that require RSC

<table>
<thead>
<tr>
<th>Under RSC (e.g., Loh &amp; Wainwright 2015)</th>
<th>Without RSC, linear regression only (e.g, Zhang et al., 2017)</th>
<th>This research on M-estimation without RSC</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O\left(\frac{1}{n}\right)$</td>
<td>$O\left(\frac{1}{\sqrt{n}}\right)$</td>
<td>$O\left(\frac{1}{n^{1/3}}\right)$</td>
</tr>
</tbody>
</table>
Theoretical Application 2: Deep Learning

**Date generation:** for some unknown $g: \mathbb{R}^d \to \mathbb{R}$

\[ b_i = g(a_i) + w_i \quad i = 1, ..., n \]

**Traditional training model:**

\[
\min_x \frac{1}{2n} \sum_{i=1}^{n} \|b_i - F_{NN}(a_i, x)\|^2
\]

- **Expressive power:** Yarotsky (2017) and Mhaskar (1996), DeVore et al. (1989), Mhaskar and Poggio (2016)

- **Two-layer, polynomial increase in dimensions, or sometimes exponential growth in layer number:** Arora et al. (2018), Barron and Klusowski (2018), Bartlett et al. (2017), Golowich et al. (2017), Neyshabur et al. (2015, 2017, 2018)

**Scarcity of theoretical guarantee on generalization performance under over-parameterization**
Approximate Sparsity and Regularization in Neural Networks

- Proposed training model: Find an $S^3$ONC solution to the formulation below

$$\min_x \frac{1}{2n} \sum_{i=1}^{n} \| b_i - F_{NN}(a_i, x) \|^2 + \sum_{j=1}^{p} P_\lambda(|x_j|)$$

- Expressive power of NN implies approximate sparsity
  - E.g., A ReLu network, with $cn^{1/3} \cdot (\ln n^{\frac{r}{3d}} + 1)$-many approximates any $r$-th order weakly-differentiable function with accuracy $O(1)n^{-\frac{r}{3d}}$ (Yarotsky, 2017)
  - $s = cn^{1/3} \cdot (\ln n^{\frac{r}{3d}} + 1)$ and $\hat{\epsilon} = O(1)n^{-\frac{r}{3d}}$ in approximate sparsity

- Empirical evidences on sparse neural networks
  - E.g., Dropout: Srivastava et al. (2014); DropConnect: Wan et al. (2013); Deep compression (remove link with small weights and retrain): Han et al. (2016)

Neural networks may not obey the RSC but they innately entail approximate sparsity
For $g: \mathbb{R}^d \to \mathbb{R}$ that has $r$-th order weak derivative;

Assume that the PDF of inputs to neurons have continuous density in a near neighborhood of 0.

$$\tilde{O}\left(\left(\frac{1}{n^3} + \frac{1}{n^6 + 6d} + \frac{1}{n^{3d}} + \sqrt{\frac{\Gamma}{n^3}}\right) \cdot \ln p + \Gamma\right)$$

Poly-logarithmic in dimensionality
For $g: \mathbb{R}^d \rightarrow \mathbb{R}$ being a polynomial function

$$\tilde{O}\left(\left(\frac{1}{n^{1/3}} + \sqrt{\frac{\Gamma}{n^{1/3}}}\right) \ln p + \Gamma\right)$$

Poly-logarithmic in dimensionality

Handwriting recognition problem:

- 28 x 28-pixel images of handwritten digits
- 60,000 training data
- 10,000 test data
**Numerical Experiment 1: MNIST**

- 800 × 800 + softmax
- Fully connected
- No preprocessing/distortion

<table>
<thead>
<tr>
<th>Model</th>
<th>Mean ± Std (%)</th>
<th>Aggregated (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN</td>
<td>1.62 ± 0.037</td>
<td>1.40</td>
</tr>
<tr>
<td>NN-Dropout</td>
<td>1.28 ± 0.040</td>
<td>1.20</td>
</tr>
<tr>
<td>DropConnect</td>
<td>1.20 ± 0.034</td>
<td>1.12</td>
</tr>
<tr>
<td>Dropout-FCP</td>
<td>1.18 ± 0.052</td>
<td>0.99</td>
</tr>
</tbody>
</table>

Wan et al., 2013, ICML
### Numerical Experiment 1: MNIST

<table>
<thead>
<tr>
<th>Layer number</th>
<th>$p$</th>
<th>NN</th>
<th>NN-Dropout</th>
<th>NN-$\ell_2$</th>
<th>NN-FCP</th>
<th>Dropout-FCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>468,500</td>
<td>2.77</td>
<td>2.86</td>
<td>3.16</td>
<td>3.00</td>
<td>2.80</td>
</tr>
<tr>
<td>3</td>
<td>470,000</td>
<td>2.64</td>
<td>2.51</td>
<td>2.87</td>
<td>2.61</td>
<td>2.55</td>
</tr>
<tr>
<td>5</td>
<td>471,600</td>
<td>3.25</td>
<td>3.02</td>
<td>2.69</td>
<td>3.26</td>
<td>2.79</td>
</tr>
<tr>
<td>9</td>
<td>472,000</td>
<td>77.32</td>
<td>3.86</td>
<td>6.36</td>
<td>3.19</td>
<td>2.83</td>
</tr>
<tr>
<td>17</td>
<td>472,800</td>
<td>30.86</td>
<td>88.65</td>
<td>88.65</td>
<td>3.89</td>
<td>3.18</td>
</tr>
</tbody>
</table>

- $500 \times 150 \times 10 \times \ldots \times 10 \times 150 + \text{softmax}$
- Fully connected
- No preprocessing/distortion
Numerical Experiment 2: Matrix Completion

$\mathbf{p} = 100 \times 100 = 10,000$, $n = 2,000$

Ratings of customers to products/restaurants/movies often constitutes an incomplete matrix
Numerical Experiment 2: Matrix Completion

<table>
<thead>
<tr>
<th>Method</th>
<th>Sub.</th>
<th>Nuclear norm (Lasso)</th>
<th>S$^3$ONC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>85.7699</td>
<td>0.1072</td>
<td>0.0005</td>
</tr>
</tbody>
</table>

\[ p = 100 \times 100 = 10,000, \ n = 2,000 \]
Summary

- The Sample Complexity analysis is an essential problem in optimization driven by data or high-dimensional/deep learning.
- Theories can be developed to match a sample size upper bound to the information theoretical lower bound, such as in RL by smart sampling.
- Based on the structure of problems/solutions, adding certain regularizations can greatly reduce the sample complexity, such as RSAA achieves poly-logarithmic sample complexity on any $S^3$ONC solution based on standard statistical and sparse assumptions.
- Although the RSAA problem become nonconvex when the regularization is concave, an $S^3$ONC solution can be efficiently computed in both theory and practice.
- Future Research: further theoretical developments and more comprehensive numerical results.