Using more data to speed-up training time

Shai Shalev-Shwartz
School of Computer Science and Engineering
The Hebrew University of Jerusalem

COST Workshop,
NIPS 2011

Based on joint work with
- Nati Srebro
- Ohad Shamir and Eran Tromer
Outline

- Time-Sample Complexity
- General Techniques:
  1. A larger hypothesis class
     - Formal Derivation of Gaps
  2. A different loss function
  3. Approximate optimization
Agnostic PAC Learning

- Domain $Z$ (e.g. $Z = \mathcal{X} \times \mathcal{Y}$)
- Hypothesis class $\mathcal{H}$ (our “inductive bias”)
- Loss function: $\ell : \mathcal{H} \times Z \rightarrow \mathbb{R}$
- $\mathcal{D}$ - unknown distribution over $Z$
- True risk: $L_D(h) = \mathbb{E}_{z \sim \mathcal{D}}[\ell(h, z)]$
- Training set: $S = (x_1, y_1), \ldots, (x_m, y_m) \overset{\text{i.i.d.}}{\sim} \mathcal{D}^m$
- Goal: use $S$ to find $h_S$ s.t.

$$\mathbb{E}_S L_D(h_S) \leq \min_{h \in \mathcal{H}} L_D(h) + \epsilon$$
Joint Time-Sample Complexity

Goal:

\[ \mathbb{E}_{S \sim D^m} [L_D(h_S)] \leq \min_{h \in \mathcal{H}} L_D(h) + \epsilon \]
Joint Time-Sample Complexity

Goal:

\[ \mathbb{E}_{S \sim D^m} [L_D(h_S)] \leq \min_{h \in \mathcal{H}} L_D(h) + \epsilon \]

- **Sample complexity**: How many examples are needed?
- **Time complexity**: How much time is needed?
Joint Time-Sample Complexity

Goal:

\[
\mathbb{E}_{S \sim D_m} [L_D(h_S)] \leq \min_{h \in \mathcal{H}} L_D(h) + \epsilon
\]

- Sample complexity: How many examples are needed?
- Time complexity: How much time is needed?

Time-sample complexity

\[ T_{\mathcal{H},\epsilon}(m) = \text{how much time is needed when } |S| = m? \]
Decatur, Goldreich, Ron 1998: “Computational Sample Complexity”
- Only distinguishes polynomial vs. non-polynomial
- Only binary classification in the realizable case
- Very few results on ”real-world” problems, e.g. Rocco Servedio showed gaps for 1-decision lists

Bottou & Bousquet 2008: “The Tradeoffs of Large Scale Learning”
- Study the effect of optimization error in generalized linear problems based on upper bounds
How Can More Data Reduce Runtime?

1. A larger hypothesis class
2. A different loss function
3. Approximate optimization
Example: Agnostic learning Preferences

The Learning Problem:

- $\mathcal{X} = [d] \times [d]$, $\mathcal{Y} = \{0, 1\}$, $Z = \mathcal{X} \times \mathcal{Y}$
- Given $(i, j) \in \mathcal{X}$ predict if $i$ is preferable over $j$
- $\mathcal{H}$ is all permutations over $[d]$
- Loss function $= \text{zero-one loss}$
Example: Agnostic learning Preferences

The Learning Problem:
- $\mathcal{X} = [d] \times [d]$, $\mathcal{Y} = \{0, 1\}$, $Z = \mathcal{X} \times \mathcal{Y}$
- Given $(i, j) \in \mathcal{X}$ predict if $i$ is preferable over $j$
- $\mathcal{H}$ is all permutations over $[d]$
- Loss function = zero-one loss

Method I:
- $\text{ERM}_{\mathcal{H}}$
- Sample complexity is $\frac{d \log(d)}{\epsilon^2}$
Example: Agnostic learning Preferences

The Learning Problem:
- \( \mathcal{X} = [d] \times [d] \), \( \mathcal{Y} = \{0, 1\} \), \( Z = \mathcal{X} \times \mathcal{Y} \)
- Given \((i, j) \in \mathcal{X}\) predict if \(i\) is preferable over \(j\)
- \(\mathcal{H}\) is all permutations over \([d]\)
- Loss function = zero-one loss

Method I:
- \(\text{ERM}_{\mathcal{H}}\)
- Sample complexity is \(\frac{d \log(d)}{\epsilon^2}\)
- Varun Kanade and Thomas Steinke (2011): If \(\text{RP} \neq \text{NP}\), it is not possible to efficiently find an \(\epsilon\)-accurate permutation
Example: Agnostic learning Preferences

The Learning Problem:

- $\mathcal{X} = [d] \times [d]$, $\mathcal{Y} = \{0, 1\}$, $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$
- Given $(i, j) \in \mathcal{X}$ predict if $i$ is preferable over $j$
- $\mathcal{H}$ is all permutations over $[d]$
- Loss function = zero-one loss

Method I:

- ERM$_{\mathcal{H}}$
- Sample complexity is $\frac{d \log(d)}{\epsilon^2}$
- Varun Kanade and Thomas Steinke (2011): If RP $\neq$ NP, it is not possible to efficiently find an $\epsilon$-accurate permutation
- Claim: If $m \geq \frac{d^2}{\epsilon^2}$ it is possible to find a predictor with error $\leq \epsilon$ in polynomial time

Varun Kanade and Thomas Steinke (2011): If RP $\neq$ NP, it is not possible to efficiently find an $\epsilon$-accurate permutation.
Example: Agnostic learning Preferences

- Let $\mathcal{H}^{(n)}$ be the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$
- $\text{ERM}_{\mathcal{H}^{(n)}}$ can be computed efficiently
- Sample complexity: $\frac{VC(\mathcal{H}^{(n)})}{\epsilon^2} = \frac{d^2}{\epsilon^2}$
- Improper learning
More Data Less Work

Using more data to speed up training time.
Lower bounds?

- Analysis is based on upper bounds
- Is it possible to (improperly) learn efficiently with $d \log(d)$ examples? (Posed as an open problem by Jake Abernathy)
- Main open problem: establish gaps by deriving lower bounds (for improper learning!)
Theorem: Assume one-way permutations exist, there exists an agnostic learning problem such that:

\[ T_{\mathcal{H},\epsilon}(m) \]

\[ 2^n + \frac{1}{\epsilon^2} \]

\[ > \text{poly}(n) \]

\[ \frac{n^3}{\epsilon^6} \]
Proof: One Way Permutations

\( P : \{0, 1\}^n \to \{0, 1\}^n \) is one-way permutation if it’s one-to-one and

- It is easy to compute \( w = P(s) \)
- It is hard to compute \( s = P^{-1}(w) \)
Proof: One Way Permutations

\[ P : \{0, 1\}^n \to \{0, 1\}^n \] is one-way permutation if it’s one-to-one and

- It is easy to compute \( w = P(s) \)
- It is hard to compute \( s = P^{-1}(w) \)

Goldreich-Levin Theorem: If \( P \) is one way, then for any algorithm \( A \),

\[ \exists w \text{ s.t. } \Pr_r[A(r, P(w)) = \langle r, w \rangle] < \frac{1}{2} + \frac{1}{\text{poly}(n)} \]
The Domain

- Let $P$ be a one-way permutation.
- $\mathcal{X} = \{0, 1\}^{2n}$, $\mathcal{Y} = \{0, 1\}$
- Domain: $Z \subset \mathcal{X} \times \mathcal{Y}$
  - $((r, s), b) \in Z$ iff $\langle P^{-1}(s), r \rangle = b$
- (Inner product over $\mathbb{GF}(2)$)
Proof: The Learning Problem

The Hypothesis Class

- $\mathcal{H} = \{ h_w : w \in \{0, 1\}^n \}$ where $h_w : \mathcal{X} \rightarrow [0, 1]$ is

$$h_w(r, s) = \begin{cases} \langle w, r \rangle & \text{if } s = P(w) \\ 1/2 & \text{o.w.} \end{cases}$$

The Loss Function:

- Absolute loss ($= \text{expected } 0-1$)

$$\ell(h, ((r, s), b)) = |h(r, s) - b|$$
Proof: The Learning Problem

The Hypothesis Class

\[ \mathcal{H} = \{ h_w : w \in \{0, 1\}^n \} \] where \( h_w : \mathcal{X} \to [0, 1] \) is

\[ h_w(r, s) = \begin{cases} \langle w, r \rangle & \text{if } s = P(w) \\ 1/2 & \text{o.w.} \end{cases} \]

The Loss Function:

- Absolute loss (= expected 0-1)

\[ \ell(h, ((r, s), b)) = |h(r, s) - b| = \begin{cases} 0 & \text{if } s = P(w) \\ 1/2 & \text{o.w.} \end{cases} \]

- Note: \( L_D(h_w) = \mathbb{P}[s \neq P(w)] \cdot \frac{1}{2} \)
Proof: The Learning Problem

The Hypothesis Class

- $\mathcal{H} = \{h_{\mathbf{w}} : \mathbf{w} \in \{0, 1\}^n\}$ where $h_{\mathbf{w}} : \mathcal{X} \rightarrow [0, 1]$ is

$$h_{\mathbf{w}}(\mathbf{r}, s) = \begin{cases} \langle \mathbf{w}, \mathbf{r} \rangle & \text{if } s = P(\mathbf{w}) \\ 1/2 & \text{o.w.} \end{cases}$$

The Loss Function:

- Absolute loss (= expected 0-1)

$$\ell(h, ((\mathbf{r}, s), b)) = |h(\mathbf{r}, s) - b| = \begin{cases} 0 & \text{if } s = P(\mathbf{w}) \\ 1/2 & \text{o.w.} \end{cases}$$

- Note: $L_D(h_{\mathbf{w}}) = \mathbb{P}[s \neq P(\mathbf{w})] \cdot \frac{1}{2}$

- Agnostic: $L_D(h_{\mathbf{w}}) = 0$ only if $\mathbb{P}[s = P(\mathbf{w})] = 1$
Proof of Second Claim

\[ 2^n + \frac{1}{\epsilon^2} > \text{poly}(n) \]

\[ \frac{n^3}{\epsilon^6} \]

Suppose we can learn with \( m = O(\log(n)) \) examples for all \( w \), define \( D_w \) s.t. \( r \) is uniform, \( s = P(w) \), and \( b = \langle r, w \rangle \).

To generate an i.i.d. training set from \( D_w \):

Pick \( r_1, \ldots, r_m \) and \( b_1, \ldots, b_m \) at random.

If \( b_i = \langle r_i, w \rangle \) for all \( i \) we're done.

This happens with probability \( \frac{1}{2^m} = \frac{1}{\text{poly}(n)} \).

Feed the training set to the learner, to get \( h_w' \approx \langle r, P(w) \rangle \).

Goldreich-Levin theorem implies a contradiction.
Proof of Second Claim

Suppose we can learn with $m = O(\log(n))$ examples

\forall w, \text{ define } D_w \text{ s.t. } r \text{ is uniform, } s = P(w), \text{ and } b = \langle r, w \rangle

To generate an i.i.d. training set from $D_w$:

- Pick $r_1, \ldots, r_m$ and $b_1, \ldots, b_m$ at random
- If $b_i = \langle r_i, w \rangle$ for all $i$ we’re done
- This happens w.p. $1/2^m = 1/poly(n)$

Feed the training set to the learner, to get $h_w'(r, P(w)) \approx \langle r, w \rangle$

Goldreich-Levin theorem $\Rightarrow$ contradiction
Proof of First Claim

Recall: $L_D(h_w) = \mathbb{P}[s \neq P(w)] \cdot \frac{1}{2} = \mathbb{P}[P^{-1}(s) \neq w] \cdot \frac{1}{2}$

Problem reduces to multiclass prediction with hypothesis class of constant predictors

Sample complexity is $1/\epsilon^2$
Proof of Third Claim

\[2^n + \frac{1}{\varepsilon^2} > \text{poly}(n)\]

\[\frac{n^3}{\varepsilon^6}\]

Original class:
\[h(w, s) = \begin{cases} \langle w, r \rangle & \text{if } s = P(w) \\ \frac{1}{2} & \text{o.w.} \end{cases}\]

New class:
\[h((r_1, s''), b_1), \ldots, (r_m', s''') = \begin{cases} \sum_i \alpha_i b_i & \text{if } r = \sum_i \alpha_i r_i \land s = s' \\ \frac{1}{2} & \text{o.w.} \end{cases}\]

New class is efficiently learnable with \(m = \frac{n}{\varepsilon^2}\)
Proof of Third Claim

\[ 2^n + \frac{1}{\epsilon^2} > \text{poly}(n) \]

\[ \frac{n^3}{\epsilon^6} \]

\[ \frac{1}{\epsilon^2}, \log(n), \frac{n}{\epsilon^2} \]

\[ m \]

\[ \mathcal{H}^{(n)} \]

Original class:
\[ h_{hw}(r, s) = \begin{cases} \langle w, r \rangle & \text{if } s = P(w) \\ 1/2 & \text{o.w.} \end{cases} \]

New class:
\[ h((r_1, s'), b_1), ..., ((r_m', s'), b_m') \]
\[ (r, s) = \sum_i \alpha_i b_i \text{ if } r = \sum_i \alpha_i r_i \land s = s' \]
\[ 1/2 \text{ o.w.} \]

New class is efficiently learnable with
\[ m = \frac{n}{\epsilon^2} \]
Proof of Third Claim

- Original class:
  \[ h_w(r,s) = \begin{cases} \langle w, r \rangle & \text{if } s = P(w) \\ 1/2 & \text{o.w.} \end{cases} \]

- New class:
  \[ h((r_1,s'), b_1),...,((r_m,s'), b_m) (r, s) = \begin{cases} \sum_i \alpha_i b_i & \text{if } r = \sum_i \alpha_i r_i \land s = s' \\ 1/2 & \text{o.w.} \end{cases} \]

- New class is efficiently learnable with \( m = n/\epsilon^2 \)
Outline

- Time-Sample Complexity ✓
- General Techniques:
  1. A larger hypothesis class ✓
     - Formal Derivation of Gaps (for a synthetic problem) ✓
  2. A different loss function
  3. Approximate optimization
Without noise, can learn efficiently even if $m = \text{sample complexity}$

With arbitrary noise, cannot learn efficiently even if $m = \infty$

(S., Shamir, Sridharan 2010)

What about stochastic noise?
Learning Margin-Based Halfspaces with Stochastic Noise

\[ \mathcal{H} = \{ x \mapsto \phi(\langle w, x \rangle) : \|w\|_2 \leq 1 \}, \quad \phi : \mathbb{R} \to [0, 1] \text{ is } \frac{1}{\mu} \text{-Lipschitz} \]

- Probabilistic classifier: \( \Pr[h_w(x) = 1] = \phi(\langle w, x \rangle) \)
- Loss function: \( \ell(w; (x, y)) = \Pr[h_w(x) \neq y] = |\phi(\langle w, x \rangle) - y| \)
- Assumption: \( \Pr[y = 1|x] = \phi(\langle w^*, x \rangle) \)
Goal: find $h$ s.t.

$$\mathbb{E}[|h(x) - y|] - \mathbb{E}[|\phi(\langle w^*, x \rangle) - y|] \leq \epsilon.$$
Learning Halfspaces with Stochastic Noise

- Goal: find $h$ s.t.
  \[
  \mathbb{E}[|h(x) - y|] - \mathbb{E}[|\phi(\langle w^*, x \rangle) - y|] \leq \epsilon .
  \]

- Idea: replace the loss function:
  \[
  \mathbb{E}[|h(x) - y|] - \mathbb{E}[|\phi(\langle w^*, x \rangle) - y|] \\
  \leq \mathbb{E}[|h(x) - \phi(\langle w^*, x \rangle)|] \\
  \leq \sqrt{\mathbb{E}[(h(x) - \phi(\langle w^*, x \rangle))^2]}
  \]
Goal: find $h$ s.t.
$$E[|h(x) - y|] - E[|\phi(\langle w^*, x \rangle) - y|] \leq \epsilon.$$ 

Idea: replace the loss function:
$$E |h(x) - y| - E |\phi(\langle w^*, x \rangle) - y|$$
$$\leq E[|h(x) - \phi(\langle w^*, x \rangle)|]$$
$$\leq \sqrt{E[(h(x) - \phi(\langle w^*, x \rangle))^2]}$$

Kalai-Sastry, Kakade-Kalai-Kanade-Shamir: The GLM-Tron algorithm learns $h$ such that
$$E[(h(x) - \phi(\langle w, x \rangle))^2] \leq O\left(\sqrt{1/\mu^2/m}\right)$$
Learning Halfspaces with Stochastic Noise

- **Goal:** Find \( h \) such that
  \[
  \mathbb{E}[|h(x) - y|] - \mathbb{E}[|\phi(\langle w^*, x \rangle) - y|] \leq \epsilon .
  \]

- **Idea:** Replace the loss function:
  \[
  \mathbb{E}|h(x) - y| - \mathbb{E}|\phi(\langle w^*, x \rangle) - y| \\
  \leq \mathbb{E}[|h(x) - \phi(\langle w^*, x \rangle)|] \\
  \leq \sqrt{\mathbb{E}[(h(x) - \phi(\langle w^*, x \rangle))^2]}
  \]

- **Kalai-Sastry, Kakade-Kalai-Kanade-Shamir:** The GLM-Tron algorithm learns \( h \) such that
  \[
  \mathbb{E}[(h(x) - \phi(\langle w, x \rangle))^2] \leq O \left( \sqrt{\frac{1/\mu^2}{m}} \right)
  \]

- **Corollary:** There is an efficient algorithm that learns Halfspaces with stochastic noise using \((1/(\mu \epsilon)^4)\) examples.
More Data Less Work

Shai Shalev-Shwartz (Hebrew U)

Using more data to speed-up training time

COST NIPS'11 22 / 30
The General Technique

\[ \mathbb{E}_S[L_D(h_S)] - \min_{h \in \mathcal{H}} L_D(h) \leq f \left( \mathbb{E}_S[L_D^{(n)}(h_S)] - \min_{h \in \mathcal{H}} L_D^{(n)}(h) \right) \]
How Can More Data Reduce Runtime?

1. A larger hypothesis class ✓
2. A different loss function ✓
3. Approximate optimization
3-term error decomposition (Bottou & Bousquet)

\[ h^* = \arg\min_{h \in \mathcal{H}} L_D(h) \quad ; \quad h^*_S = \arg\min_{h \in \mathcal{H}} L_S(h) \]
3-term error decomposition (Bottou & Bousquet)

\[ h^* = \arg\min_{h \in \mathcal{H}} L_D(h) \quad ; \quad h_{S}^* = \arg\min_{h \in \mathcal{H}} L_S(h) \]

\[ L_D(h_S) = L_D(h^*) + L_D(h_S^*) - L_D(h^*) + L_D(h_S) - L_D(h_S^*) \]

approximation \quad estimation \quad optimization
3-term error decomposition (Bottou & Bousquet)

\[ h^* = \arg\min_{h \in \mathcal{H}} L_D(h) \quad ; \quad h^*_S = \arg\min_{h \in \mathcal{H}} L_S(h) \]

\[ L_D(h_S) = \underbrace{L_D(h^*)}_{\text{approximation}} + \underbrace{L_D(h^*_S) - L_D(h^*)}_{\text{estimation}} + \underbrace{L_D(h_S) - L_D(h^*_S)}_{\text{optimization}} \]
Convex Learning Problems

- $\mathcal{H}$ is a convex set
- For all $z$, the function $\ell(\cdot, z)$ is convex and Lipschitz
- Example: SVM learning (hinge-loss minimization)
Solving Convex Learning Problems

Goal: \( \min_h \mathbb{E}_{z \sim D}[\ell(h, z)] \)

\[
\min_h \frac{1}{m} \sum_{i=1}^{m} \ell(h, z_i)
\]

Both methods have the same sample complexity in the worst case. But, ERM can be better on many distributions.

Shai Shalev-Shwartz (Hebrew U)

Using more data to speed-up training time

COST NIPS'11 27 / 30
Solving Convex Learning Problems

Goal: \( \min_h \mathbb{E}_{z \sim D}[\ell(h, z)] \)

ERM

\( \min_h \frac{1}{m} \sum_{i=1}^{m} \ell(h, z_i) \)

SGD

\( h \leftarrow h - \eta \nabla \ell(h, z_i) \)

- Both methods have the same sample complexity in the worst case.
- But, ERM can be better on many distributions
Second-Order Stochastic Gradient Descent

- Smaller sample complexity under some spectral assumptions, E.g. Leon Bottou’s talk today
- But, runtime is $\Omega(d^2)$ per iteration
- When $d$ is large, we might prefer running SGD (for approximately solving the ERM problem)
More Data Less Work for SGD

Theoretical

Empirical (CCAT)

Training Set Size

Million Iterations (≈ runtime)

Shai Shalev-Shwartz (Hebrew U)

Using more data to speed-up training time

COST NIPS'11 29 / 30
Summary

- A formal model for Time-Sample Complexity
- Different techniques for improving training time when more examples are available
- Formal derivation of gaps

Open Questions

- Other techniques?
- Showing gaps for real-world problems?