Stochastic Approximation:
Mini-Batches, Optimistic Rates
and Acceleration

Nati Srebro
Toyota Technological Institute at Chicago
a philanthropically endowed academic computer science institute
dedicated to basic research and graduate education in computer science

Joint work with:
Andy Cotter (TTIC), Karthik Sridharan (TTIC→UPenn),
Ohad Shamir (Microsoft), Ambuj Tewari (TTIC→Austin)
Outline

- Learning
- Mini-Batches
- “Optimistic Rates”
- Acceleration
Optimization for Learning

- Empirical Risk Minimization:

\[
\min_w \frac{1}{m} \sum_{i=1}^{m} \ell(w, (x_i, y_i))
\]

\[\ell(w, (x, y)) = \text{loss}(\langle w, x \rangle, y)\]
Optimization for Learning

• Empirical Risk Minimization:

$$\min_{w} \frac{1}{m} \sum_{i=1}^{m} \ell(w, (x_i, y_i)) + \lambda \|w\|^2$$

$$\ell(w, (x, y)) = \text{loss}(\langle w, x \rangle, y)$$
Optimization for Learning

• **Empirical Risk Minimization:**

\[
\min_{\|w\| \leq B} \frac{1}{m} \sum_{i=1}^{m} \ell(w, (x_i, y_i))
\]

\[
\ell(w, (x, y)) = \text{loss}(<w, x>, y)
\]
Optimization for Learning

- Empirical Risk Minimization:
  \[
  \min_{\|w\| \leq B} \frac{1}{m} \sum_{i=1}^{m} \ell(w, (x_i, y_i))
  \]
  \[
  \ell(w, (x, y)) = \text{loss}((w, x), y)
  \]

- SGD:
  \[
  w^+ \leftarrow \Pi_B (w - \eta \nabla_w \ell(w, (x_i, y_i)))
  \[
  i \sim \text{Unif}[1..m] \text{ picked at random at each iteration}
Learning is Optimization!

$$\min_w L(w)$$

$$L(w) = \mathbb{E}_{(x,y) \sim World}[\ell(w, (x, y))]$$
Learning is Optimization!

\[
\begin{align*}
\min_w L(w) \\
L(w) &= \mathbb{E} \, z \sim_{World} [\ell(w, z)]
\end{align*}
\]
Learning is Optimization!

\[
\min_w L(w)
\]

\[
L(w) = \mathbb{E}_{z \sim \text{World}}[\ell(w, z)]
\]

**SAA/ERM**

sample \( z_1, \ldots, z_m \sim D \) then:

\[
\min_{\|w\| \leq B} \frac{1}{m} \sum_{i=1}^{m} \ell(w, z_i)
\]

**SA**

\[
w^+ \leftarrow \Pi_B(w - \eta \nabla_w \ell(w, z_i))
\]
Learning is Optimization!

\[
\min_w L(w)
\]

\[
L(w) = \mathbb{E}_{z \sim \text{World}}[\ell(w, z)]
\]

**SA (Stochastic Approximation)**

\[
w^+ \leftarrow w - \eta \nabla_w \ell(w, z)
\]

**SAA/ERM**

Sample \( z_1, \ldots, z_m \sim D \) then:

\[
\min_{\|w\| \leq B} \frac{1}{m} \sum_{i=1}^{m} \ell(w, z_i)
\]

SGD Guarantee gives Learning Guarantee:

\[
L(\overline{w}^{(k)}) \leq L(w^*) + \sqrt{\frac{\|w^*\|^2 R^2}{k}}
\]

\( k = m = \text{#iteration} = \text{#samples} \)

\[||\nabla \ell|| \leq R, \quad \text{e.g. for } (w, \ell(x,y)) = \text{loss}(\langle w, x \rangle, y), \ |\text{loss'}| \leq 1: \ ||x|| \leq R \]
Outline

• Learning
• **Mini-Batches**
• “Optimistic Rates”
• Acceleration
Stochastic Gradient Descent

- Parallelization
  - actually need $b >> \#\text{machines}$ due to communication overhead

\[
g(k) = \frac{1}{b} \sum_{i=1}^{b} \nabla \ell(w^{(k)}, z_{k,i})
\]

\[
w^{(k+1)} \leftarrow w^{(k)} - \eta^{(k)} g^{(k)}
\]

- Reduce overhead (loop control, function calls, etc)

- $g = \frac{1}{3} \sum \nabla \ell(w, z_i)$

- $b = \text{minibatch size}$

- $k = \#\text{iter} = \text{parallel runtime}$

- $m = k \cdot b = \#\text{samples}$

- $\eta^{(k)} = \text{learning rate}$

- $w \leftarrow w - \frac{1}{3} \sum \nabla \ell(w, z_i)$

- $z_1, z_2, \ldots, z_{12}$

- $w \leftarrow w - \nabla \ell(w, z_1)$

- $z_{13}$

- $w \leftarrow w - \frac{1}{3} \sum \nabla \ell(w, z_i)$

- $w \leftarrow w - \nabla \ell(w, z_2)$

- $z_{14}$

- $w \leftarrow w - \frac{1}{3} \sum \nabla \ell(w, z_i)$

- $w \leftarrow w - \nabla \ell(w, z_3)$

- $z_{15}$

- $w \leftarrow w - \frac{1}{3} \sum \nabla \ell(w, z_i)$

- $w \leftarrow w - \nabla \ell(w, z_4)$

- $z_{16}$

- $w \leftarrow w - \frac{1}{3} \sum \nabla \ell(w, z_i)$

- $w \leftarrow w - \nabla \ell(w, z_5)$

- $z_{17}$

- $w \leftarrow w - \frac{1}{3} \sum \nabla \ell(w, z_i)$

- $w \leftarrow w - \nabla \ell(w, z_6)$

- $z_{18}$

- $w \leftarrow w - \frac{1}{3} \sum \nabla \ell(w, z_i)$

- $w \leftarrow w - \nabla \ell(w, z_7)$

- $z_{19}$

- $w \leftarrow w - \frac{1}{3} \sum \nabla \ell(w, z_i)$

- $w \leftarrow w - \nabla \ell(w, z_8)$

- $z_{20}$

- $w \leftarrow w - \frac{1}{3} \sum \nabla \ell(w, z_i)$

- $w \leftarrow w - \nabla \ell(w, z_9)$

- $z_{21}$

- $w \leftarrow w - \frac{1}{3} \sum \nabla \ell(w, z_i)$

- $w \leftarrow w - \nabla \ell(w, z_{10})$
Stochastic Gradient Descent

- Parallelization
  - actually need $b >> \#machines$ due to communication overhead
- Reduce overhead (loop control, function calls, etc)
- If projection expensive: reduce $\#$projections
  - e.g. $\|W\|_{tr} \leq B$
Stochastic Gradient Descent

\[
g^{(k)} = \frac{1}{b} \sum_{i=1}^{b} \nabla \ell (w^{(k)}, z_{k,i})
\]

\[
w^{(k+1)} \leftarrow w^{(k)} - \eta^{(k)} g^{(k)}
\]

- Parallelization
  - actually need \( b >> \# \text{machines} \) due to communication overhead
- Reduce overhead (loop control, function calls, etc)
- If projection expensive: reduce \# projections
  - e.g. \( ||W||_{tr} \leq B \)
- We don’t expect gain in terms of pure “sequential runtime” \( m \)
Using Mini-Batches I

\[ w^{(k+1)} \leftarrow w^{(k)} - \eta^{(k)} g^{(k)} \]

\[ g^{(k)} = \frac{1}{b} \sum_{i=1}^{b} \nabla \ell \left( w^{(k)}, z_{k,i} \right) \]

• For convex (non-smooth) loss (with \( R \leq 1 \)):

\[ \text{Var}(g^{(k)}) = \frac{R^2}{b} \Rightarrow L(\bar{w}^{(k)}) \leq L(w^*) + \sqrt{\frac{\|w^*\|^2}{k}} + \sqrt{\frac{\|w^*\|^2}{kb}} \]

• But for smooth loss (i.e. \(|\text{loss''}| \leq 1\), \( L(w) \) has Lip. grad):

\[ L(\bar{w}^{(k)}) \leq L(w^*) + O \left( \frac{\|w^*\|^2}{\sqrt{kb}} + \frac{\|w^*\|^2}{k} \right) \]

\( \Rightarrow \) Linear speedup (no sequential slow-down) until:

\[ b = k = \sqrt{m} \]
Outline

• Learning
• Mini-Batches
• “Optimistic Rates”
• Acceleration
Optimistic Rates

- For smooth, **non-negative** $\ell(w,z)$ (with $b=1$, i.e. $m=k$):

  
  \[ L(w^{(m)}) \leq L(w^*) + O \left( \sqrt{\frac{\|w^*\|^2 R^2 L(w^*)}{m}} + \frac{\|w^*\|^2 R^2}{m} \right) \]

  and this is best possible with $m$ samples.

- Follows from self-bounding property— for **non-negative** $f(w)$ with H-Lip gradient:

  \[ \|f(w)\| \leq \sqrt{4Hf(w)} \]

- Sample (=iteration) complexity:

  \[ k = m = O \left( \frac{\|w^*\| R^2}{\epsilon} \left( \frac{L^* + \epsilon}{\epsilon} \right) \right) \]
Optimistic Rates with Mini-Batches

\[ w^{(k+1)} \leftarrow w^{(k)} - \eta^{(k)} g^{(k)}, \quad g^{(k)} = \frac{1}{b} \sum_{i=1}^{b} \nabla \ell \left( w^{(k)} , z_{k,i} \right) \]

- For smooth non-negative loss, with \( L^* = L(w^*) \) (and \( R \leq 1 \)):

\[
L(\overline{w}^{(k)}) \leq L(w^*) + O \left( \sqrt{\frac{\|w^*\|^2}{kb} L^*} + \frac{\|w^*\|^2}{kb} + \frac{\|w^*\|^2}{k} \right)
\]

[Cotter Shamir S Sridharan 11]
Optimistic Rates with Mini-Batches

\[ \mathbf{w}^{(k+1)} \leftarrow \mathbf{w}^{(k)} - \eta^{(k)} \mathbf{g}^{(k)}, \quad \mathbf{g}^{(k)} = \frac{1}{b} \sum_{i=1}^{b} \nabla \ell \left( \mathbf{w}^{(k)}, z_{k,i} \right) \]

- For smooth non-negative loss, with \( L^* = L(\mathbf{w}^*) \) (and \( R \leq 1 \)):

\[
L(\mathbf{w}^{(k)}) \leq L(\mathbf{w}^*) + O \left( \sqrt{\frac{\| \mathbf{w}^* \|^2 L^*}{kb}} + \frac{\| \mathbf{w}^* \|^2}{k} \right)
\]

\[ \Rightarrow \quad k = O \left( \frac{\| \mathbf{w}^* \|^2}{\epsilon} \left( \frac{L^*/b + \epsilon}{\epsilon} \right) \right) \]

\[ \Rightarrow \text{no speedup (ie linear sequential slowdown) past} \]

\[ b = L^*/\epsilon \]
Outline

• Learning
• Mini-Batches
• “Optimistic Rates”
• Acceleration
Acceleration

- Recall dependence on number of samples:

\[
L(w^{(m)}) \leq L(w^*) + O \left( \sqrt{\frac{\|w^*\|^2 R^2 L(w^*)}{m}} + \frac{\|w^*\|^2 R^2}{m} \right)
\]

- Getting $1/k$ is not enough for speedup when 2nd term is dominant, but using acceleration can get $1/k^2$.

- Could thus hope to get:

\[
L(w^{(k)}) \leq L(w^*) + O \left( \sqrt{\frac{\|w^*\|^2 L^*}{kb}} + \frac{\|w^*\|^2}{kb} + \frac{\|w^*\|^2}{k^2} \right)
\]
Accelerated Mini-Batch Descent

\[ u = \beta_k^{-1} v^{(k)} + (1 - \beta_k^{-1}) w^{(k)} \]

\[ g = \frac{1}{b} \sum_{i=1}^{b} \nabla \ell(u, z_{k,i}) \]

\[ v^{(k+1)} \leftarrow \Pi_B(u - \gamma_i g) \]

\[ w^{(k+1)} \leftarrow \beta_k^{-1} v^{(k+1)} + (1 - \beta_k^{-1}) w^{(k)} \]

\[ \beta_k = \frac{k + 1}{2} \]

\[ \gamma_k = \gamma_0 k^p \]

\[ \gamma = \min \left\{ \frac{1}{4}, \sqrt{\frac{b B^2}{412 L^* (k-1)^{2p+1}}}, \left(\frac{b}{1044 (k-1)^{2p}}\right)^{\frac{p+1}{2p+1}} \left(\frac{B^2}{4 B^2 + \sqrt{4 B^2 L^*}}\right)^{\frac{p}{2p+1}} \right\} \]

\[ p = \min \left\{ \max \left\{ \frac{\log(b)}{2 \log(k-1)}, \frac{\log \log(k)}{2 \log(b(k-1)) - \log \log(k)} \right\}, 1 \right\} \]

For a non-negative smooth loss and \( \|w^*\| \leq B \):

\[ L(w^{(k)}) \leq L(w^*) + \tilde{O} \left( \sqrt{\frac{B^2 L^*}{kb}} + \frac{B^2}{k \sqrt{b}} + \frac{B^2 \log(k)}{kb} + \frac{B^2}{k^2} \right) \]

[Cotter Shamir S Sridharan 11]
Accelerated Mini-Batch Descent

\[ u = \beta_k^{-1} v^{(k)} + (1 - \beta_k^{-1}) w^{(k)} \]
\[ g = \frac{1}{b} \sum_{i=1}^{b} \nabla \ell(u, z_{k,i}) \]
\[ v^{(k+1)} \leftarrow \Pi_B(u - \gamma_i g) \]
\[ w^{(k+1)} \leftarrow \beta_k^{-1} v^{(k+1)} + (1 - \beta_k^{-1}) w^{(k)} \]
\[ \beta_k = \frac{k + 1}{2} \]
\[ \gamma_k = \gamma_0 k^p \]
\[ \gamma = \min \left\{ \frac{1}{4}, \sqrt{\frac{bB^2}{412L^*(k-1)^{2p+1}}}, \left( \frac{b}{1044(k-1)^{2p}} \right)^{p+1} \left( \frac{B^2}{4B^2 + \sqrt{4B^2L^*}} \right)^{\frac{p}{2p+1}} \right\} \]
\[ p = \min \left\{ \max \left\{ \frac{\log(b)}{2\log(k-1)}, \frac{\log \log(k)}{2(\log(b(k-1))) - \log \log(k)} \right\}, 1 \right\} \]

For a non-negative smooth loss and \( ||w^*|| \leq B \):

\[ L(w^{(k)}) \leq L(w^*) + \tilde{O} \left( \sqrt{\frac{B^2 L^*}{kb}} + \frac{B^2}{k \sqrt{b}} + \frac{B^2}{k^2} \right) \]

\( \Rightarrow \) even if \( L^* = O(\epsilon) \), still get \( b^{1/2} \) speedup until:

\[ b = k^2 = m^{2/3} \]
Experiments: dependence on $p$

$$\gamma_k = \gamma_0 k^p$$

Ruieters CCAT $L^*=0$, $m=18578$, optimal value of $\gamma_0$
Experiments

CoverType

Rutgers CCAT

$L^*=0$
Summary

- Mini-batches useful but tricky
- Acceleration helps, even essential theoretically

Open issues:
- Our upper bound:
  \[ L(w^{(k)}) \leq L(w^*) + O \left( \sqrt{\frac{B^2L^*}{kb}} + \frac{B^2}{k\sqrt{b}} + \frac{B^2\log(k)}{kb} + \frac{B^2}{k^2} \right) \]
  is it possible to get:
  \[ L(w^{(k)}) \leq L(w^*) + O \left( \sqrt{\frac{B^2L^*}{kb}} + \frac{B^2}{kb} + \frac{B^2}{k^2} \right) \]
  and is projection really necessary?
- Is it enough to require \( L(w) \) is smooth (even if \( \ell(w) \) is not)?

- Srebro, Sridharan, Tewair, *Smoothness, Low Noise and Fast Rate*, NIPS’10