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:

$$\min_{\mathbf{w}} \;\; \frac{1}{m} \sum_{i=1}^{m} \ell(\mathbf{w}, (\mathbf{x}_i, y_i))$$

$$\ell(w, (x, y)) = \mathrm{loss}(\langle \mathbf{w}, \mathbf{x} \rangle, y)$$

Optimization for Learning

- Empirical Risk Minimization:

$$\min_{\mathbf{w}} \quad \frac{1}{m}\sum_{i=1}^{m} \ell(\mathbf{w},(\mathbf{x}_i, y_i)) + \lambda \left\|\mathbf{w}\right\|^2$$

$$\ell(w,(x,y)) = \mathrm{loss}(\langle \mathbf{w}, \mathbf{x}\rangle, y)$$

Optimization for Learning

- Empirical Risk Minimization:

$$\min_{\|\mathbf{w}\| \leq B} \frac{1}{m} \sum_{i=1}^{m} \ell(\mathbf{w}, (\mathbf{x}_i, y_i))$$

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

Optimization for Learning

- Empirical Risk Minimization:

$$\min_{\|\mathbf{w}\| \leq B} \frac{1}{m} \sum_{i=1}^{m} \ell(\mathbf{w}, (\mathbf{x}_i, y_i))$$

$$\ell(w, (x, y)) = \mathrm{loss}(\langle \mathbf{w}, \mathbf{x} \rangle, y)$$

- SGD:

$$\mathbf{w}^{+} \leftarrow \Pi_B \left(\mathbf{w} - \eta \nabla_{\mathbf{w}} \ell(\mathbf{w}, (\mathbf{x}_i, y_i)) \right)$$

$i \sim$ Unif[1..m] picked at random at each iteration

Learning is Optimization!

$$\min_{\mathbf{w}} L(\mathbf{w})$$

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

Learning is Optimization!

$$\min_{\mathbf{w}} L(\mathbf{w})$$

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

Learning is Optimization!

$$\min_{\mathbf{w}} L(\mathbf{w})$$

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

$\Downarrow$ SAA/ERM

sample $z_1, \ldots z_m \sim \mathcal{D}$ then:

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

$\Downarrow$ SA

$$\mathbf{w}^{+} \leftarrow \Pi_B(\mathbf{w} - \eta \nabla_{\mathbf{w}} \ell(\mathbf{w}, z_i))$$

Learning is Optimization!

$$\min_{\mathbf{w}} L(\mathbf{w})$$

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

SA (Stochastic Approximation)

SAA/ERM

$$\mathbf{w}^+ \leftarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} \ell(\mathbf{w}, z)$$

$z \sim World$

sample $z_1,\ldots z_m \sim \mathcal{D}$ then:

$$\min_{\|\mathbf{w}\| \le B} \frac{1}{m} \sum_{i=1}^{m} \ell(\mathbf{w}, z_i)$$

SA

$$\mathbf{w}^+ \leftarrow \Pi_B(\mathbf{w} - \eta \nabla_{\mathbf{w}} \ell(\mathbf{w}, z_i))$$

SGD Guarantee gives Learning Guarantee:

$$L(\overline{\mathbf{w}}^{(k)}) \le L(\mathbf{w}^*) + \sqrt{\frac{\|\mathbf{w}^*\|^2 R^2}{k}}$$

k = m = #iteration = #samples

$\|\nabla \ell\| \le R$,

e.g. for $(w, \ell(x,y)) = loss(\langle w,x \rangle, y)$, $|loss'| \le 1$:

$\|x\| \le R$

Outline

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

Stochastic Gradient Descent

$\mathbf{w} \leftarrow \mathbf{w} - \nabla\ell(w, z_1)$	z_1	
$\mathbf{w} \leftarrow \mathbf{w} - \nabla\ell(w, z_2)$	z_2	$g = \frac{1}{3}\sum \nabla\ell(\mathbf{w}, z_i)$
$\mathbf{w} \leftarrow \mathbf{w} - \nabla\ell(w, z_3)$	z_3	$\mathbf{w} \leftarrow \mathbf{w} - \frac{1}{3}\sum \nabla\ell(\mathbf{w}, z_i)$
$\mathbf{w} \leftarrow \mathbf{w} - \nabla\ell(w, z_4)$	z_4	
$\mathbf{w} \leftarrow \mathbf{w} - \nabla\ell(w, z_5)$	z_5	$g = \frac{1}{3}\sum \nabla\ell(\mathbf{w}, z_i)$
$\mathbf{w} \leftarrow \mathbf{w} - \nabla\ell(w, z_6)$	z_6	$\mathbf{w} \leftarrow \mathbf{w} - \frac{1}{3}\sum \nabla\ell(\mathbf{w}, z_i)$
$\mathbf{w} \leftarrow \mathbf{w} - \nabla\ell(w, z_7)$	z_7	
$\mathbf{w} \leftarrow \mathbf{w} - \nabla\ell(w, z_8)$	z_8	$g = \frac{1}{3}\sum \nabla\ell(\mathbf{w}, z_i)$
$\mathbf{w} \leftarrow \mathbf{w} - \nabla\ell(w, z_9)$	z_9	$\mathbf{w} \leftarrow \mathbf{w} - \frac{1}{3}\sum \nabla\ell(\mathbf{w}, z_i)$
$\mathbf{w} \leftarrow \mathbf{w} - \nabla\ell(w, z_{10})$	z_{10}	
$\mathbf{w} \leftarrow \mathbf{w} - \nabla\ell(w, z_{11})$	z_{11}	$g = \frac{1}{3}\sum \nabla\ell(\mathbf{w}, z_i)$
$\mathbf{w} \leftarrow \mathbf{w} - \nabla\ell(w, z_{12})$	z_{12}	$\mathbf{w} \leftarrow \mathbf{w} - \frac{1}{3}\sum \nabla\ell(\mathbf{w}, z_i)$

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

$$\mathbf{w}^{(k+1)} \leftarrow \mathbf{w}^{(k)} - \eta^{(k)}\mathbf{g}^{(k)}$$

b = minibatch size

k = #iter = parallel runtime

m = k·b = #samples
= sequential runtime

- Parallelization
  - actually need b>>#machines due to communication overhead
- Reduce overhead (loop control, function calls, etc)

Stochastic Gradient Descent

$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_1)$ | z_1
$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_2)$ | z_2
$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_3)$ | z_3
$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_4)$ | z_4
$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_5)$ | z_5
$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_6)$ | z_6
$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_7)$ | z_7
$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_8)$ | z_8
$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_9)$ | z_9
$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_{10})$ | z_{10}
$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_{11})$ | z_{11}
$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_{12})$ | z_{12}

$\left.\begin{matrix} z_1 \\ z_2 \\ z_3 \end{matrix}\right\} g = \frac{1}{3} \sum \nabla \ell(\mathbf{w}, z_i)$
$\mathbf{w} \leftarrow \mathbf{w} - \frac{1}{3} \sum \nabla \ell(\mathbf{w}, z_i)$

$\left.\begin{matrix} z_4 \\ z_5 \\ z_6 \end{matrix}\right\} g = \frac{1}{3} \sum \nabla \ell(\mathbf{w}, z_i)$
$\mathbf{w} \leftarrow \mathbf{w} - \frac{1}{3} \sum \nabla \ell(\mathbf{w}, z_i)$

$\left.\begin{matrix} z_7 \\ z_8 \\ z_9 \end{matrix}\right\} g = \frac{1}{3} \sum \nabla \ell(\mathbf{w}, z_i)$
$\mathbf{w} \leftarrow \mathbf{w} - \frac{1}{3} \sum \nabla \ell(\mathbf{w}, z_i)$

$\left.\begin{matrix} z_{10} \\ z_{11} \\ z_{12} \end{matrix}\right\} g = \frac{1}{3} \sum \nabla \ell(\mathbf{w}, z_i)$
$\mathbf{w} \leftarrow \mathbf{w} - \frac{1}{3} \sum \nabla \ell(\mathbf{w}, z_i)$

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

$$\mathbf{w}^{(k+1)} \leftarrow \Pi(\mathbf{w}^{(k)} - \eta^{(k)} \mathbf{g}^{(k)})$$

b = minibatch size

k = #iter = parallel runtime

m = k·b = #samples
= sequential runtime

- 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

$$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_1)$$
$$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_2)$$
$$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_3)$$
$$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_4)$$
$$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_5)$$
$$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_6)$$
$$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_7)$$
$$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_8)$$
$$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_9)$$
$$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_{10})$$
$$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_{11})$$
$$\mathbf{w} \leftarrow \mathbf{w} - \nabla \ell(w, z_{12})$$

z_1
z_2
z_3
z_4
z_5
z_6
z_7
z_8
z_9
z_{10}
z_{11}
z_{12}

$$\left.\begin{matrix} z_1 \\ z_2 \\ z_3 \end{matrix}\right\} g = \tfrac{1}{3} \sum \nabla \ell(\mathbf{w}, z_i) \qquad \mathbf{w} \leftarrow \mathbf{w} - \tfrac{1}{3} \sum \nabla \ell(\mathbf{w}, z_i)$$
$$\left.\begin{matrix} z_4 \\ z_5 \\ z_6 \end{matrix}\right\} g = \tfrac{1}{3} \sum \nabla \ell(\mathbf{w}, z_i) \qquad \mathbf{w} \leftarrow \mathbf{w} - \tfrac{1}{3} \sum \nabla \ell(\mathbf{w}, z_i)$$
$$\left.\begin{matrix} z_7 \\ z_8 \\ z_9 \end{matrix}\right\} g = \tfrac{1}{3} \sum \nabla \ell(\mathbf{w}, z_i) \qquad \mathbf{w} \leftarrow \mathbf{w} - \tfrac{1}{3} \sum \nabla \ell(\mathbf{w}, z_i)$$
$$\left.\begin{matrix} z_{10} \\ z_{11} \\ z_{12} \end{matrix}\right\} g = \tfrac{1}{3} \sum \nabla \ell(\mathbf{w}, z_i) \qquad \mathbf{w} \leftarrow \mathbf{w} - \tfrac{1}{3} \sum \nabla \ell(\mathbf{w}, z_i)$$

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

$$\mathbf{w}^{(k+1)} \leftarrow \mathbf{w}^{(k)} - \eta^{(k)} \mathbf{g}^{(k)}$$

b = minibatch size

k = #iter = parallel runtime

m = k·b = #samples
= sequential runtime

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

Using Mini-Batches I

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

- For convex (non-smooth) loss (with R≤1):

$$\mathrm{Var}(\mathbf{g}^{(k)}) = \frac{R^2}{b} \quad \Rightarrow \quad L(\overline{\mathbf{w}}^{(k)}) \leq L(\mathbf{w}^*) + \sqrt{\frac{\|\mathbf{w}^*\|^2}{k}} + \sqrt{\frac{\|\mathbf{w}^*\|^2}{kb}}$$

- But for smooth loss (i.e. |loss''|≤1, L(w) has Lip. grad):

[Lan 09][Dekel et al 10]

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

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

$$b = k = \sqrt{m}$$

Outline

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

Optimistic Rates

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

[S **Sridharan Tewari 10**]

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

and this is best possible with m samples.

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

$$\|f(\mathbf{w})\| \leq \sqrt{4Hf(\mathbf{w})}$$

- Sample (=iteration) complexity:

$$k = m = O\left(\frac{\|\mathbf{w}^*\| R^2}{\epsilon}\left(\frac{L^* + \epsilon}{\epsilon}\right)\right)$$

Optimistic Rates with Mini-Batches

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

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

[Cotter Shamir S **Sridharan 11]**

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

Optimistic Rates with Mini-Batches

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

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

[Cotter Shamir S **Sridharan 11]**

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

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

$\Rightarrow$ no speedup (ie linear sequential slowdown) past $b=L^*/\epsilon$

Outline

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

Acceleration

- Recall dependence on number of samples:

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

- 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:

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

Accelerated Mini-Batch Descent

$$\mathbf{u} = \beta_k^{-1}\mathbf{v}^{(k)} + (1-\beta_k^{-1})\mathbf{w}^{(k)}$$

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

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

$$\mathbf{w}^{(k+1)} \leftarrow \beta_k^{-1}\mathbf{v}^{(k+1)} + (1-\beta_k^{-1})\mathbf{w}^{(k)}$$

$$\beta_k = \frac{k+1}{2}$$

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

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

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

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

[Cotter Shamir S **Sridharan 11]**

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

Accelerated Mini-Batch Descent

$$\boxed{\begin{aligned}
\mathbf{u} &= \beta_k^{-1}\mathbf{v}^{(k)} + (1-\beta_k^{-1})\mathbf{w}^{(k)} \\
\mathbf{g} &= \frac{1}{b}\sum_{i=1}^{b} \nabla\ell(\mathbf{u}, z_{k,i}) \\
\mathbf{v}^{(k+1)} &\leftarrow \Pi_B(\mathbf{u} - \gamma_i \mathbf{g}) \\
\mathbf{w}^{(k+1)} &\leftarrow \beta_k^{-1}\mathbf{v}^{(k+1)} + (1-\beta_k^{-1})\mathbf{w}^{(k)}
\end{aligned}}$$

$$\beta_k = \frac{k+1}{2}$$

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

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

$$p = \min\left\{\max\left\{\tfrac{\log(b)}{2\log(k-1)}, \tfrac{\log\log(k)}{2(\log(b(k-1))-\log\log(k))}\right\}, 1\right\}$$

For a non-negative smooth loss and ||w*|| $\leq$ B:

[Cotter Shamir S **Sridharan 11]**

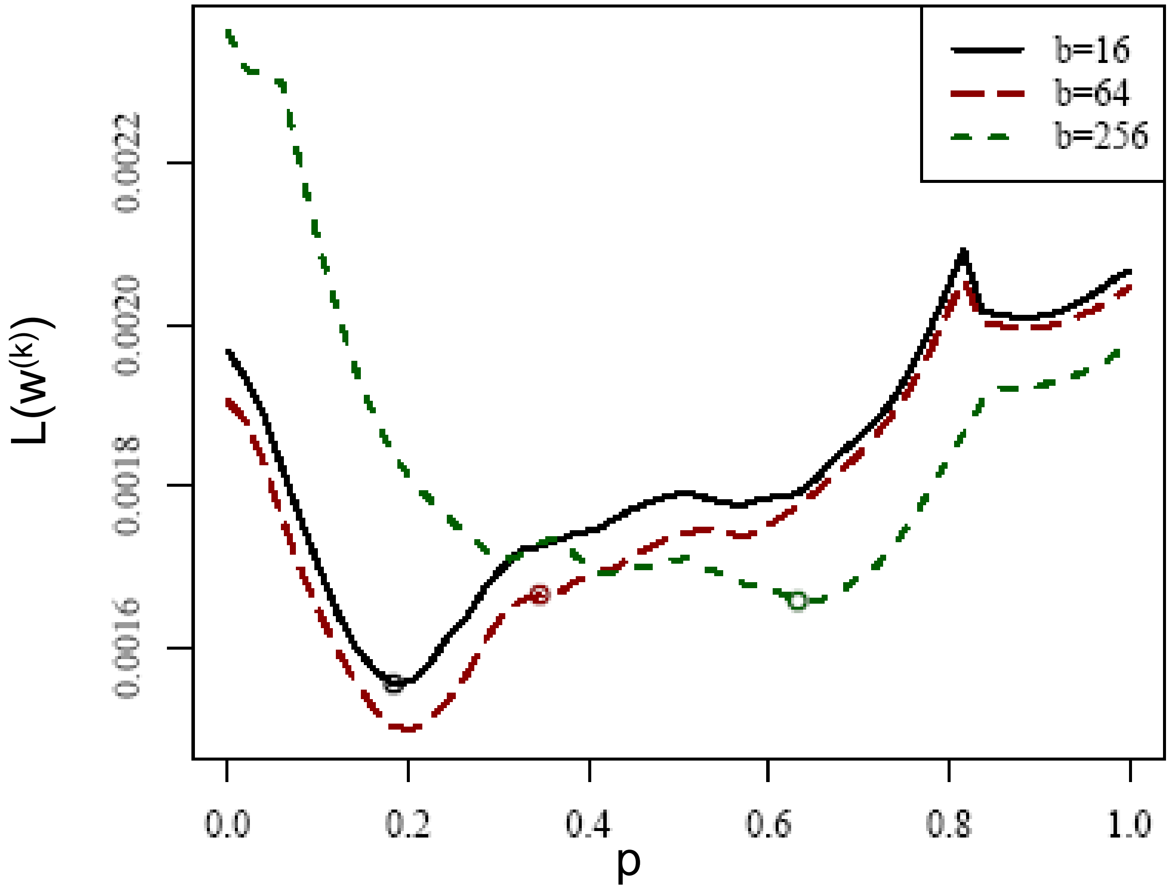
$$L(\mathbf{w}^{(k)}) \leq L(\mathbf{w}^*) + \tilde{O}\left(\sqrt{\frac{B^2L^*}{kb}} + \frac{B^2}{k\sqrt{b}} + \frac{B^2}{k^2}\right)$$

$\Rightarrow$ even if L* = O(ϵ), still get b½ speedup until:

b = k² = m$^{2/3}$

Experiments: dependence on p

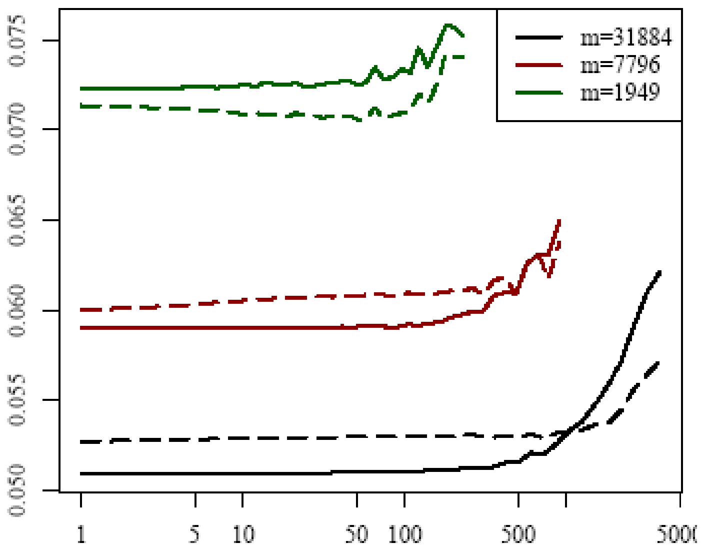
$$\gamma_k = \gamma_0 k^p$$



Rueters CCAT $L^*=0$, m=18578, optimal value of γ_o

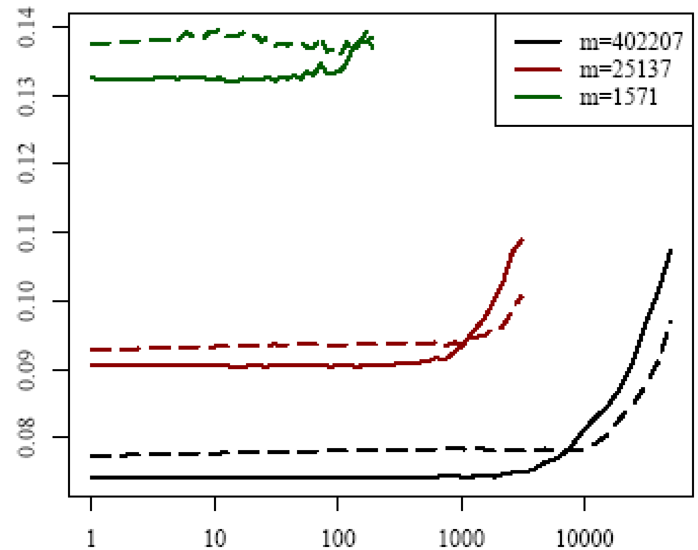
Experiments

CoverType



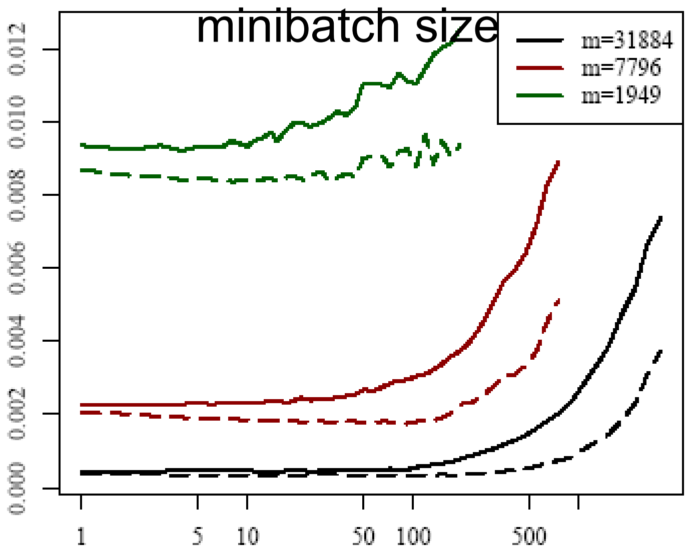
minibatch size

Rueters CCAT

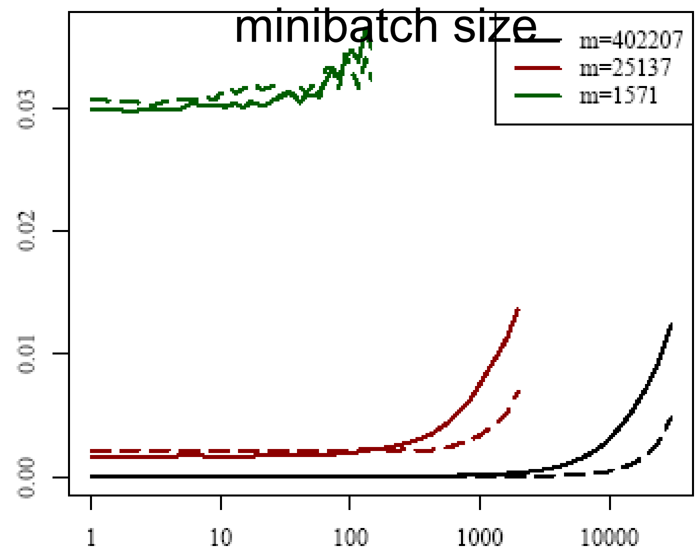


minibatch size

L*=0



minibatch size



minibatch size

Summary

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

Open issues:

- Our upper bound:

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

  is it possible to get:

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

  and is projection really necessary?
- Is it enough to require L(w) is smooth (even if ℓ(w) is not)?

- Srebro, Sridharan, Tewair, **Smoothness, Low Noise and Fast Rate**, NIPS'10
- Cotter, Shamir, Srebro, Sridharan, **Better Mini-Batch Algorithms via Accelerated Gradient Methods**, NIPS'11