Optimal Distributed Online Prediction using Mini-Batches

Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao

Microsoft Research

NIPS Workshop on Learning on Cores, Clusters and Clouds
December 11, 2010
Motivation

• online algorithms often studied in serial setting
  – fast, simple, good generalization, . . .
  – but *sequential* in nature
Motivation

- online algorithms often studied in serial setting
  - fast, simple, good generalization, . . .
  - but *sequential* in nature
- web-scale online prediction (e.g., search engines)
  - inputs arrive at *high rate*
  - need to provide *real-time* service

critical to use parallel/distributed computing
Motivation

• online algorithms often studied in serial setting
  – fast, simple, good generalization, . . .
  – but *sequential* in nature

• web-scale online prediction (e.g., search engines)
  – inputs arrive at *high rate*
  – need to provide *real-time* service
    critical to use parallel/distributed computing

• how well can online algorithms (old or new) perform in distributed setting?
Stochastic online prediction

- repeat for each $i = 1, 2, 3, \ldots$
  - predict $w_i \in \mathcal{W}$ (e.g., based on $\nabla f(w_{i-1}, z_{i-1})$)
  - receive $z_i$ drawn i.i.d. from fixed distribution
  - suffer loss $f(w_i, z_i)$
Stochastic online prediction

- repeat for each \( i = 1, 2, 3, \ldots \)
  - predict \( w_i \in \mathcal{W} \) (e.g., based on \( \nabla f(w_{i-1}, z_{i-1}) \))
  - receive \( z_i \) drawn i.i.d. from fixed distribution
  - suffer loss \( f(w_i, z_i) \)

- measure quality of predictions using regret

\[
R(m) = \sum_{i=1}^{m} (f(w_i, z_i) - f(w^*, z_i))
\]

- \( w^* = \arg \min_{w \in \mathcal{W}} \mathbb{E}_z[f(w, z)] \)
- assume \( f(\cdot, z) \) convex, \( \mathcal{W} \) closed and convex
Stochastic optimization

• find approximate solution to

\[
\begin{align*}
\text{minimize} \quad & F(w) \triangleq \mathbb{E}_z[f(w, z)] \\
\text{subject to} \quad & w \in W
\end{align*}
\]

• success measured by \textit{optimality gap}

\[
G(m) = F(w_m) - F(w^*)
\]

• different motivations
  – often used to solve large-scale batch problem
  – usually no real-time requirement

• how can parallel computing speed up solution?
Distributed online prediction

- system has $k$ nodes
- network model
  - limited bandwidth
  - latency
  - non-blocking
- measure same regret

$$R(m) = \sum_{i=1}^{m} \left( f\left( w_i, z_i \right) - f\left( w^*, z_i \right) \right)$$
Limits of performance

- an ideal (but unrealistic) solution
  - run serial algorithm on a “super” computer that is $k$ times faster
  - optimal regret bound: $\mathbb{E}[R(m)] \leq O(\sqrt{m})$
Limits of performance

- an ideal (but unrealistic) solution
  - run serial algorithm on a “super” computer that is $k$ times faster
  - optimal regret bound: $\mathbb{E}[R(m)] \leq O(\sqrt{m})$

- a trivial (no-communication) solution
  - each node operates in isolation
  - regret bound scales poorly with network size $k$

\[
\mathbb{E}[R(m)] \leq k \cdot O\left(\sqrt{\frac{m}{k}}\right) = O\left(\sqrt{km}\right)
\]
Related work and contribution

- previous work on distributed optimization
  - Tsitsiklis, Bertsekas and Athans (1986); Tsitsiklis and Bertsekas (1989); Nedić, Bertsekas and Bokar (2001); Nedić and Ozdaglar (2009); ...
  - Langford, Smola and Zinkevich (2009); Duchi, Agarwal and Wainwright (2010); Zinkevich, Weimar, Smola and Li (2010); ...
Related work and contribution

- previous work on distributed optimization
  - Tsitsiklis, Bertsekas and Athans (1986); Tsitsiklis and Bertsekas (1989); Nedić, Bertsekas and Bokar (2001); Nedić and Ozdaglar (2009); …
  - Langford, Smola and Zinkevich (2009); Duchi, Agarwal and Wainwright (2010); Zinkevich, Weimar, Smola and Li (2010); …

- when applied to problems considered here

<table>
<thead>
<tr>
<th>Trivial</th>
<th>Ideal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Online prediction</td>
<td>$O(\sqrt{km})$</td>
</tr>
<tr>
<td>Stochastic optimization</td>
<td>$O\left(\frac{1}{\sqrt{T}}\right)$</td>
</tr>
</tbody>
</table>
Related work and contribution

• previous work on distributed optimization
  – Tsitsiklis, Bertsekas and Athans (1986); Tsitsiklis and Bertsekas (1989); Nedić, Bertsekas and Bokar (2001); Nedić and Ozdaglar (2009); 
  – Langford, Smola and Zinkevich (2009); Duchi, Agarwal and Wainwright (2010); Zinkevich, Weimar, Smola and Li (2010); 

• when applied to problems considered here

\[
\text{online prediction: } O(\sqrt{km}) \quad \text{stochastic optimization: } O\left(\frac{1}{\sqrt{T}}\right) \\
\text{our results: } O(\sqrt{m}) \quad O\left(\frac{1}{\sqrt{kT}}\right)
\]
Outline

- motivation and introduction
- variance bounds for serial algorithms
- DMB algorithm and regret bounds
- parallel stochastic optimization
- experiments on a web-scale problem
Serial online algorithms

- projected gradient descent
  \[ w_{j+1} = \pi_W \left( w_j - \frac{1}{\alpha_j} g_j \right) \]

- dual averaging method
  \[ w_{j+1} = \arg \min_{w \in W} \left\{ \left\langle \sum_{i=1}^{j} g_i, w \right\rangle + \alpha_j h(w) \right\} \]
Serial online algorithms

- projected gradient descent
  \[ w_{j+1} = \pi_W \left( w_j - \frac{1}{\alpha_j} g_j \right) \]

- dual averaging method
  \[ w_{j+1} = \arg \min_{w \in W} \left\{ \left\langle \sum_{i=1}^{j} g_i, w \right\rangle + \alpha_j h(w) \right\} \]

optimal regret bound (attained by \( \alpha_j = \Theta(\sqrt{j}) \)):
\[ \mathbb{E}[R(m)] = O(\sqrt{m}) \]
Variance bounds

- additional assumptions
  - smoothness: \( \forall z \in Z, \forall w, w' \in W, \)
    \[
    \| \nabla_w f(w, z) - \nabla_w f(w', z) \| \leq L \| w - w' \|
    \]
  - bounded gradient variance: \( \forall w \in W, \)
    \[
    \mathbb{E}_z \left[ \| \nabla_w f(w, z) - \nabla F(w) \|^2 \right] \leq \sigma^2
    \]
Variance bounds

• additional assumptions
  – smoothness: \( \forall z \in Z, \forall w, w' \in W, \)
  \[
  \| \nabla_w f(w, z) - \nabla_w f(w', z) \| \leq L \| w - w' \|
  \]
  – bounded gradient variance: \( \forall w \in W, \)
  \[
  \mathbb{E}_z \left[ \| \nabla_w f(w, z) - \nabla F(w) \| \right]^2 \leq \sigma^2
  \]

• **Theorem:** refined bound using \( \alpha_j = L + (\sigma / D) \sqrt{j} \)
  \[
  \mathbb{E}[R(m)] \leq 2D^2L + 2D\sigma\sqrt{m}
  \]
Variance bounds

- additional assumptions
  - smoothness: \( \forall z \in Z, \forall w, w' \in W, \)
  \[
  \| \nabla_w f(w, z) - \nabla_w f(w', z) \| \leq L \| w - w' \|
  \]
  - bounded gradient variance: \( \forall w \in W, \)
  \[
  \mathbb{E}_z \left[ \| \nabla_w f(w, z) - \nabla F(w) \| \right]^2 \leq \sigma^2
  \]

- **Theorem:** refined bound using \( \alpha_j = L + (\sigma / D) \sqrt{j} \)
  \[
  \mathbb{E}[R(m)] \leq 2D^2 L + 2D \sigma \sqrt{m} \triangleq \psi(\sigma^2, m)
  \]
Variance reduction via mini-batching

- mini-batching
  - predict $b$ samples using same predictor
  - update predictor based on average gradients

not a new idea, but no theoretical support
Variance reduction via mini-batching

- mini-batching
  - predict $b$ samples using same predictor
  - update predictor based on average gradients

_not a new idea, but no theoretical support_

- our analysis: consider averaged cost function

\[
\bar{f}(w, (z_1, \ldots, z_b)) \triangleq \frac{1}{b} \sum_{s=1}^{b} f(w, z_s)
\]

- $\nabla_w \bar{f}$ has variance $\frac{\sigma^2}{b}$; at most $\left\lceil \frac{m}{b} \right\rceil$ batches
- serial regret bound:

\[
b \cdot \psi\left(\frac{\sigma^2}{b}, \left\lceil \frac{m}{b} \right\rceil\right) \leq 2bD^2L + 2D\sigma\sqrt{m + b}
\]
Distributed mini-batch (DMB)

- for each node
  - accumulate gradients of first $b/k$ inputs
  - vector-sum to compute $\bar{g}_j$ over $b$ gradients
  - update $w_{j+1}$ based on $\bar{g}_j$
Distributed mini-batch (DMB)

- for each node
  - accumulate gradients of first $b/k$ inputs
  - vector-sum to compute $\bar{g}_j$ over $b$ gradients
  - update $w_{j+1}$ based on $\bar{g}_j$

- expected regret bound

$$(b + \mu) \psi \left( \frac{\sigma^2}{b}, \left[ \frac{m}{b + \mu} \right] \right)$$
Regret bound for DMB

• suppose $\psi(\sigma^2, m) = 2D^2 L + 2D \sigma \sqrt{m}$
  
  – if $b = m^\rho$ for any $\rho \in (0, 1/2)$, then
    \[ \mathbb{E}[R(m)] \leq 2D \sigma \sqrt{m} + o(\sqrt{m}) \]

  – choose $b = m^{1/3}$, bound becomes
    \[ 2D \sigma \sqrt{m} + 2D (LD + \sigma \sqrt{\mu}) m^{1/3} + O(m^{1/6}) \]

• asymptotically optimal: dominant term same as in ideal serial solution
Regret bound for DMB

• suppose $\psi(\sigma^2, m) = 2D^2 L + 2D\sigma \sqrt{m}$
  
  – if $b = m^\rho$ for any $\rho \in (0, 1/2)$, then
    $$\mathbb{E}[R(m)] \leq 2D\sigma \sqrt{m} + o(\sqrt{m})$$
  
  – choose $b = m^{1/3}$, bound becomes
    $$2D\sigma \sqrt{m} + 2D (LD + \sigma \sqrt{\mu}) m^{1/3} + O(m^{1/6})$$

• asymptotically optimal: dominant term same as in ideal serial solution

• scale nicely with latency: often $\mu \propto \log(k)$
Stochastic Optimization

- find approximate solution to
  \[
  \minimize_{w \in W} F(w) \triangleq \mathbb{E}_z[f(w, z)]
  \]
  
- success measured by optimality gap
  \[
  G(m) = F(\bar{w}_m) - F(w^*)
  \]
  
- for convex loss and i.i.d. inputs
  \[
  \mathbb{E}[G(m)] \leq \frac{1}{m} \mathbb{E}[R(m)] \leq \frac{1}{m} \psi(\sigma^2, m)
  \]
Stochastic Optimization

• find approximate solution to

\[
\min_{w \in W} F(w) \triangleq \mathbb{E}_z[f(w, z)]
\]

• success measured by optimality gap

\[
G(m) = F(\tilde{w}_m) - F(w^*)
\]

• for convex loss and i.i.d. inputs

\[
\mathbb{E}[G(m)] \leq \frac{1}{m} \mathbb{E}[R(m)] \leq \frac{1}{m} \psi(\sigma^2, m) \triangleq \bar{\psi}(\sigma^2, m)
\]
DMB for stochastic optimization

- for each node
  - accumulate gradients of $b/k$ inputs
  - vector-sum to compute $\bar{g}_j$ over $b$ gradients
  - update $w_{j+1}$ based on $\bar{g}_j$
DMB for stochastic optimization

• for each node
  – accumulate gradients of $b/k$ inputs
  – vector-sum to compute $\bar{g}_j$ over $b$ gradients
  – update $w_{j+1}$ based on $\bar{g}_j$

• bound on optimality gap

$$\mathbb{E}[G(m)] \leq \bar{\psi}\left(\frac{\sigma^2}{b}, \frac{m}{b}\right)$$
DMB for stochastic optimization

• if serial gap is \( \bar{\psi}(\sigma^2, m) = \frac{2D^2L}{m} + \frac{2D\sigma}{\sqrt{m}} \), then

\[
\mathbb{E}[G(m)] \leq \bar{\psi} \left( \frac{\sigma^2}{b}, \frac{m}{b} \right) = \frac{2bD^2L}{m} + \frac{2D\sigma}{\sqrt{m}}
\]
DMB for stochastic optimization

- if serial gap is \( \bar{\psi}(\sigma^2, m) = \frac{2D^2L}{m} + \frac{2D\sigma}{\sqrt{m}} \), then

\[
\mathbb{E}[G(m)] \leq \bar{\psi}\left(\frac{\sigma^2}{b}, \frac{m}{b}\right) = \frac{2bD^2L}{m} + \frac{2D\sigma}{\sqrt{m}}
\]

- parallel speed-up

\[
S = \frac{m}{\frac{m}{b} \left(\frac{b}{k} + \delta\right)} = \frac{k}{1 + \frac{\delta}{b}k}
\]

- asymptotic linear speed-up with \( b \propto m^{1/3} \)
- similar result for reaching same optimality gap
Web-scale experiments

- an online binary prediction problem
  - predict *highly monetizable* queries
  - log of $10^9$ queries issued to a commercial search engine

- logistic loss function

\[ f(w, z) = \log(1 + \exp(-\langle w, z \rangle)) \]

- algorithm: stochastic dual averaging method
  (separate $5 \times 10^8$ queries for parameter tuning)
Experiments: serial mini-batching

![Graph showing the relationship between average loss and number of inputs for different batch sizes (b=1, b=32, b=1024).](chart.png)

- Average loss decreases as the number of inputs increases.
- Different batch sizes result in different rates of decrease.

<table>
<thead>
<tr>
<th>Batch Size</th>
<th>Number of Inputs</th>
<th>Average Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>b=1</td>
<td>10^5</td>
<td>0.85</td>
</tr>
<tr>
<td>b=32</td>
<td>10^6</td>
<td>0.75</td>
</tr>
<tr>
<td>b=1024</td>
<td>10^7</td>
<td>0.65</td>
</tr>
</tbody>
</table>
Experiments: DMB vs. others

$k=1024, \mu=40, b=1024$

The graph compares the average loss of different methods as the number of inputs increases. The methods include:

- **no-comm** (dashed red line)
- **batch no-comm** (dotted cyan line)
- **serial** (dotted black line)
- **DMB** (solid blue line)

The graph shows how each method performs under varying numbers of inputs, with DMB generally maintaining a lower average loss compared to the others.
Experiments: DMB vs. others

$k=32$, $\mu=20$, $b=1024$

![Graph showing comparison between DMB and other methods across different datasets.](image)
Experiments: effects of latency

b=1024

average loss

number of inputs

\( \mu = 40 \)
\( \mu = 320 \)
\( \mu = 1280 \)
\( \mu = 5120 \)
Experiments: optimal batch size

- fixed cluster size $k = 32$ (latency $\mu = 20$)
- empirical observations
  - large batch size $(b = 512)$ beneficial at first
  - small batch size $(b = 128)$ better in the end
Summary

- distributed stochastic online prediction
  - DMB turns serial algorithms into parallel ones
  - optimal $O(\sqrt{m})$ regret bound for smooth loss
- stochastic optimization: near linear speed-up
- *first* provable demonstration that distributed computing worthwhile for these two problems
Summary

- distributed stochastic online prediction
  - DMB turns serial algorithms into parallel ones
  - optimal $O(\sqrt{m})$ regret bound for smooth loss
- stochastic optimization: near linear speed-up
- first provable demonstration that distributed computing worthwhile for these two problems

Future directions

- DMB in asynchronous distributed environment (progress made, report available on arXiv)
- non-smooth functions? non-stochastic inputs?