Sub-Sampled Newton Methods for Machine Learning

Jorge Nocedal
Northwestern University

Goldman Lecture, Sept 2016
Collaborators

Raghu Bollapragada
Northwestern University

Richard Byrd
University of Colorado
Dramatic breakdown of barriers between optimization and statistics. Partly stimulated by the rise of machine learning.

Learning process and learning measures defined using continuous optimization models – convex and non-convex.

Nonlinear optimization algorithms used to train statistical models – a great challenge with the advent of high dimensional models and huge data sets.

The stochastic gradient method currently plays a central role.
1. Why has it risen to such prominence?

2. What is the main mechanism that drives it?

3. What can we say about its behavior in convex and non-convex cases?

4. What ideas have been proposed to improve upon SG?

``Optimization Methods for Machine Learning'',
Bottou, Curtis, Nocedal (2016)
Problem statement

Given training set \( \{(x_1, y_1), \ldots (x_n, y_n)\} \)

Given a loss function \( \ell(z, y) \)  

(hinge loss, logistic,...) 

Find a prediction function \( h(x; w) \)  

(linear, DNN,...) 

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

Notation: \( f_i(w) = \ell(h(x_i; w), y_i) \)

\[
R_n(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w) \\
\text{empirical risk}
\]

Random variable \( \xi = (x, y) \)

\[
F(w) = \mathbb{E}[f(w; \xi)] \\
\text{expected risk}
\]
Stochastic Gradient Method

First present algorithms for empirical risk minimization

\[ R_n(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w) \]

\[ w_{k+1} = w_k - \alpha_k \nabla f_i(w_k) \quad i \in \{1, \ldots, n\} \text{ choose at random} \]

• Very cheap, noisy iteration; gradient w.r.t. just 1 data point
• Not a gradient descent method
• Stochastic process dependent on the choice of \( i \)
• Descent in expectation
Batch Optimization Methods

\[ w_{k+1} = w_k - \alpha_k \nabla R_n(w_k) \]

batch gradient method

\[ w_{k+1} = w_k - \frac{\alpha_k}{n} \sum_{i=1}^{n} \nabla f_i(w_k) \]

- More expensive, accurate step
- Can choose among a wide range of optimization algorithms
- Opportunities for parallelism

Why has SG emerged as the preeminent method?

Computational trade-offs between stochastic and batch methods

Ability to minimize \( F \) (generalization error)
Practical Experience

Logistic regression; speech data

Fast initial progress of SG followed by drastic slowdown

Can we explain this?

10 epochs
Intuition

SG employs information more efficiently than batch methods

**Argument 1:**
Suppose data consists of 10 copies of a set $S$
Iteration of batch method 10 times more expensive
SG performs same computations
Computational complexity

Total work to obtain $R_n(w_k) \leq R_n(w^*) + \epsilon$

<table>
<thead>
<tr>
<th>Batch gradient method:</th>
<th>$n \log(1/\epsilon)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stochastic gradient method:</td>
<td>$1/\epsilon$</td>
</tr>
</tbody>
</table>

Think of $\epsilon = 10^{-3}$

Which one is better?

More precisely:
Batch: $nd\kappa \log(1/\epsilon)$
SG: $d\nu\kappa^2 / \epsilon$
Example by Bertsekas

\[ R_n(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w) \]

Note that this is a **geographical** argument

**Analysis:** given \( w_k \) what is the **expected decrease** in the objective function \( R_n \) as we choose one of the quadratics randomly?
A fundamental observation

\[ \mathbb{E}[R_n(w_{k+1}) - R_n(w_k)] \leq -\alpha_k \| \nabla R_n(w_k) \|^2 + \alpha_k^2 \mathbb{E} \| \nabla f_{i_k}(w_k) \|^2 \]

Initially, gradient decrease dominates; then variance in gradient hinders progress.

To ensure convergence: \( \alpha_k \to 0 \) in SG method to control variance. Sub-sampled Newton methods directly control the noise given in the last term.
What can we say when \( \alpha_k = \alpha \) is constant?
Theorem 4.6 (Strongly Convex Objective, Fixed Stepsize). Under Assumptions 4.1, 4.3, and 4.5 (with $F_{\text{inf}} = F_*$), suppose that the SG method (Algorithm 4.1) is run with a fixed stepsize, $\alpha_k = \bar{\alpha}$ for all $k \in \mathbb{N}$, satisfying

$$0 < \bar{\alpha} \leq \frac{\mu}{LM_G}.$$  \hfill (4.13)

Then, for all $k \in \mathbb{N}$ the expected optimality gap satisfies:

$$E[F(w_k) - F_*] \leq \frac{\bar{\alpha}LM}{2c\mu} + (1 - \bar{\alpha}c\mu)^{k-1} \left(F(w_1) - F_* - \frac{\bar{\alpha}LM}{2c\mu}\right)$$

$$\xrightarrow{k \to \infty} \frac{\bar{\alpha}LM}{2c\mu}.$$  \hfill (4.14)

- Only converges to a neighborhood of the optimal value.
If we wait long enough, halving the stepsize \( \alpha \) eventually halves \( F(\mathbf{w} \downarrow k) - F \uparrow \star \).

We can even estimate \( F \uparrow \star \approx 2F \downarrow \alpha / 2 - F \downarrow \alpha \).
Although much is understood about the SG method, still some great mysteries: why is it so much better than batch methods on DNNs?
Sharp and flat minima

Keskar et al. (2016)

Observing $R$ along line
From SG solution to batch solution
Goodfellow et al

Deep convolutional
Neural net CIFAR-10

SG solution
Batch solution

**SG**: mini-batch of size 256

**Batch**: 10% of training set

ADAM optimizer
Testing accuracy and sharpness

Keskar (2016)

Sharpness of minimizer vs batch size

**Sharpness:**

Max $R$ in a small box around minimizer

Testing accuracy vs batch size
Drawback of SG method: distributed computing

SG is notoriously hard to parallelize
- Because it updates the parameters $\nu$ with high frequency
- Because it slows down with delayed updates.

SG still works with relaxed synchronization
- Because this is just a little bit more noise.

Communication overhead give room for new opportunities
- There is ample time to compute things while communication takes place.
- Opportunity for optimization algorithms with higher per-iteration costs
 ➔ SG may not be the best algorithm for distributed training.
Sub-sampled Newton Methods
Iteration

\[ R_n(w) = \frac{1}{n} \sum_{i=1}^{n} f_i(w) \]

Choose \( S \subset \{1,\ldots,n\}, \quad X \in \{1,\ldots,n\} \) uniformly and independently

\[
\nabla^2 F_S(w_k) p = -\nabla F_X(w_k) \quad w_{k+1} = w_k + \alpha_k p
\]

Sub-sampled gradient and Hessian

\[
\nabla F_X(w_k) = \frac{1}{|X|} \sum_{i \in X} \nabla f_i(w_k) \quad \nabla^2 F_S(w_k) = \frac{1}{|S|} \sum_{i \in S} \nabla^2 f_i(w_k)
\]

True Newton method impractical in large-scale machine learning
Will not achieve scale invariance or quadratic convergence
But the stochastic nature of the objective creates opportunities:
Coordinate Hessian sample \( S \) and gradient sample \( X \) for optimal complexity
Active research area

- Friedlander and Schmidt (2011)
- Royset (2012)
- Erdogdu and Montanari (2015)
- Roosta-Khorasani and Mahoney (2016)
- Agarwal, Bullins and Hazan (2016)
- Pilanci and Wainwright (2015)
- Pasupathy, Glynn, Ghosh, Hashemi (2015)
- Xu, Yang, Roosta-Khorasani, Re’, Mahoney (2016)
Linear convergence

\[ \nabla^2 F_{S_k}(w_k)p = -\nabla F_{X_k}(w_k) \quad w_{k+1} = w_k + \alpha p \]

The following result is well known for strongly convex objective:

**Theorem:** Under standard assumptions. If

a) \( \alpha = \frac{\mu}{L} \)

b) \( |S_k| = \text{constant} \)

c) \( |X_k| = \eta^k \quad \eta > 1 \) (geometric growth)

Then, \( \mathbb{E}[\|w_k - w^*\|] \rightarrow 0 \) at a linear rate and work complexity matches that of stochastic gradient method

\( \mu = \) smallest eigenvalue of any subsampled Hessian

\( L = \) largest eigenvalue of Hessian of \( F \)
Local superlinear convergence  

Objective: expected risk $F$

We can show the linear-quadratic result

$$
\mathbb{E}_k[\| w_{k+1} - w^* \|] \leq C_1 \| w_k - w^* \|^2 + \frac{\sigma \| w_k - w^* \|}{\mu \sqrt{|S_k|}} + \frac{\nu}{\mu \sqrt{|X_k|}}
$$

To obtain superlinear convergence:

i) $|S_k| \rightarrow \infty$

ii) $|X_k|$ must increase faster than geometrically
Closer look at the constant

We can show the linear-quadratic result

\[ \mathbb{E}_k[\| w_{k+1} - w^* \|] \leq C_1 \| w_k - w^* \|^2 + \frac{\sigma \| w_k - w^* \|}{\mu \sqrt{|S_k|}} + \frac{\nu}{\mu \sqrt{|X_k|}} \]

\[ \| \mathbb{E}[(\nabla^2 F_i(w) - \nabla^2 F(w))^2] \| \leq \sigma^2 \quad \text{tr}(\text{Cov}(\nabla f_i(w))) \leq \nu^2 \]
Achieving faster convergence

We can show the linear-quadratic result

$$
E_k[\|w_{k+1} - w^*\|] \leq C_1 \|w_k - w^*\|^2 + \frac{\sigma \|w_k - w^*\|}{\mu \sqrt{|S_k|}} + \frac{\nu}{\mu \sqrt{|X_k|}}
$$

To obtain we used the bound

$$
E_k[\|\nabla^2 F_{S_k}(w_k) - \nabla^2 F(w_k)\| \leq \frac{\sigma}{\sqrt{|S_k|}} \|w_k - w^*\|}
$$

Not matrix concentration inequalities
Formal superlinear convergence

Theorem: under the conditions just stated, there is a neighborhood of $w^*$ such that for the sub-sampled Newton method with $\alpha_k = 1$

$$\mathbb{E}[\| w_k - w^* \|] \to 0 \quad \text{superlinearly}$$
Observations on quasi-Newton methods

\[ B_k p = -\nabla F_x(w_k) \quad w_{k+1} = w_k + \alpha_k p \quad \text{and suppose } w_k \to w^* \]

If
\[
\frac{\| [B_k - \nabla^2 f(x^*)] p_k \|}{\| p_k \|} \to 0 \quad \text{convergence rate is superlinear}
\]

1. \( B_k \) need not converge to \( B^* \); only good approximation along search directions
Hacker’s definition of second-order method

1. One for which unit steplength is acceptable (yields sufficient reduction) most of the time

2. Why? How can one learn the scale of search directions without having learned curvature of the function in some relevant spaces?

3. This ``definition” is problem dependent
Inexact Methods

\[ \nabla^2 F_S(w_k) p = -\nabla F_X(w_k) \quad w_{k+1} = w_k + \alpha_k p \]

1. Exact method: Hessian approximation is inverted (e.g. Newton-sketch)
2. Inexact method: solve linear system inexactly by iterative solver
3. Conjugate gradient
4. Stochastic gradient
5. Both require only Hessian-vector products

Newton-CG chooses a fixed sample \( S \), applies CG to

\[ q_k(p) = F(w_k) + \nabla F(w_k)^T p + \frac{1}{2} p^T \nabla^2 F_S(w_k) p \]
Newton-SGI (stochastic gradient iteration)

If we apply the standard gradient method to

\[ q_k(p) = F(w_k) + \nabla F(w_k)^T p + \frac{1}{2} p^T \nabla^2 F(w_k) p \]

we obtain the iteration

\[ p_{k+1}^i = p_k^i - \nabla q_k(p_k^i) = (I - \nabla^2 F(w_k)) p_k^i - \nabla F(w_k) \]

Consider instead the semi-stochastic gradient iteration:

1. Choose and index \( j \) at random;

2. \( p_{k+1}^i = (I - \nabla^2 F_j(w_k)) p_k^i - \nabla F(w_k) \)

Gradient method using the exact gradient but and an estimate of the Hessian. This method is implicit in Agarwal, Bullins, Hazan 2016

Change sample Hessian at each inner iteration
Comparing Newton-CG and Newton-GD

Number of Hessian-vector products to achieve

\[ \| w_{k+1} - w^* \| \leq \frac{1}{2} \| w_k - w^* \| \quad (\ast) \]

\[
O\left( \left( \hat{\kappa}_l^{\text{max}} \right)^2 \hat{\kappa}_l \log(\hat{\kappa}_l) \log(d) \right) \quad \text{Newton-SGI}
\]

\[
O\left( \left( \hat{\kappa}_l^{\text{max}} \right)^2 \sqrt{\hat{\kappa}_l^{\text{max}}} \log(\hat{\kappa}_l^{\text{max}}) \right) \quad \text{Newton-CG}
\]

Results in Agarwal, Bullins and Hazan (2016) and Xu, Yang, Re, Roosta-Khorasani, Mahoney (2016):

Decrease (\ast) obtained at each step with probability 1-p
Our results give convergence of the whole sequence in expectation
Complexity bounds are very pessimistic, particularly for CG
Final Remarks

- The search for effective optimization algorithms for machine learning is ongoing
- In spite of the total dominance of the SG method at present on very large scale applications
- SG does not parallelize well
- SG is a first order method affected by ill conditioning
- A method that is noisy at the start and gradually becomes more accurate seems attractive