Exploiting Fast Local Convergence of Second-Order Methods Globally: Adaptive Sample Size Methods

Aryan Mokhtari

The University of Texas at Austin

SIAM Conference on Computational Science and Engineering (CSE21)
Beyond First-Order Methods in Machine Learning Mini-symposium
March 3rd, 2021
Data: we are given $N$ observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, for $i = 1, \ldots, N$
Data: we are given $N$ observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, for $i = 1, \ldots, N$

Prediction function: $h(x, w)$ parametrized by $w \in \mathbb{R}^d$

⇒ Linear model: $h(x, w) = w^\top x$

⇒ Neural Nets: $h(x, w) = w_3^\top \sigma(w_2^\top \sigma(w_1^\top x))$
Data: we are given $N$ observations $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$, for $i = 1, \ldots, N$

Prediction function: $h(x, w)$ parametrized by $w \in \mathbb{R}^d$

- Linear model: $h(x, w) = w^\top x$
- Neural Nets: $h(x, w) = w_3^\top \sigma(w_2^\top \sigma(w_1^\top x))$

Loss function: $\ell : \mathcal{Y} \to \mathbb{R}$ measures the prediction error $\ell(h(x, w), y)$

- Quadratic loss: $\ell(h(x, w), y) = \frac{1}{2}(h(x, w) - y)^2$
- Logistic loss: $\ell(h(x, w), y) = \log(1 + \exp(-yh(x, w)))$
Parametric Supervised Machine Learning

- **Data:** we are given $N$ observations $(x_i, y_i) \in X \times Y$, for $i = 1, \ldots, N$

- **Prediction function:** $h(x, w)$ parametrized by $w \in \mathbb{R}^d$
  - Linear model: $h(x, w) = w^\top x$
  - Neural Nets: $h(x, w) = w_3^\top \sigma(w_2^\top \sigma(w_1^\top x))$

- **Loss function:** $\ell : Y \rightarrow \mathbb{R}$ measures the prediction error $\ell(h(x, w), y)$
  - Quadratic loss: $\ell(h(x, w), y) = \frac{1}{2}(h(x, w) - y)^2$
  - Logistic loss: $\ell(h(x, w), y) = \log(1 + \exp(-yh(x, w)))$

- The objective function of Empirical Risk Minimization (ERM)
  $$\min_w \frac{1}{N} \sum_{i=1}^{N} \ell(h(x_i, w), y_i)$$
Many Machine Learning tasks can be formulated as the described problem:
Many Machine Learning tasks can be formulated as the described problem:

- Click through rate prediction in Ad clicking for search engines
Many Machine Learning tasks can be formulated as the described problem:

- Click through rate prediction in Ad clicking for search engines
- Face recognition
Data driven classification

Many Machine Learning tasks can be formulated as the described problem:

- Click through rate prediction in Ad clicking for search engines
- Face recognition
- Recommender systems
We talked about empirical risk minimization (ERM)

\[ w_N^* := \arg\min_{w \in \mathbb{R}^p} R_N(w) := \arg\min_{w \in \mathbb{R}^p} \frac{1}{N} \sum_{i=1}^{N} \ell(h(x_i, w), y_i) \]
Back to the ERM problem

- We talked about empirical risk minimization (ERM)

\[ w_N^* := \arg \min_{w \in \mathbb{R}^p} R_N(w) := \arg \min_{w \in \mathbb{R}^p} \frac{1}{N} \sum_{i=1}^{N} \ell(h(x_i, w), y_i) \]

- Why is it called “empirical”?

⇒ Empirical distribution to approximate the underlying data distribution
Back to the ERM problem

▶ We talked about empirical risk minimization (ERM)

\[ w_N^* := \arg\min_{w \in \mathbb{R}^p} R_N(w) := \arg\min_{w \in \mathbb{R}^p} \frac{1}{N} \sum_{i=1}^{N} \ell(h(x_i, w), y_i) \]

▶ Why is it called “empirical”?

⇒ Empirical distribution to approximate the underlying data distribution

▶ Our original goal is to solve the expected risk minimization problem

\[ w^* := \arg\min_{w \in \mathbb{R}^p} R(w) := \arg\min_{w \in \mathbb{R}^p} \mathbb{E}_{x,y}[\ell(h(x, w), y)] \]
Back to the ERM problem

- We talked about empirical risk minimization (ERM)

\[
\mathbf{w}_N^* := \arg\min_{\mathbf{w} \in \mathbb{R}^p} R_N(\mathbf{w}) := \arg\min_{\mathbf{w} \in \mathbb{R}^p} \frac{1}{N} \sum_{i=1}^{N} \ell(h(x_i, \mathbf{w}), y_i)
\]

- Why is it called “empirical”?
  \[\Rightarrow\] Empirical distribution to approximate the underlying data distribution

- Our original goal is to solve the expected risk minimization problem

\[
\mathbf{w}^* := \arg\min_{\mathbf{w} \in \mathbb{R}^p} R(\mathbf{w}) := \arg\min_{\mathbf{w} \in \mathbb{R}^p} \mathbb{E}_{x,y}[\ell(h(x, \mathbf{w}), y)]
\]

- ERM approximates the main problem \[\Rightarrow\] No need for perfect solution
From statistical learning literature we know that

\[ \sup_w |R(w) - R_N(w)| \leq V_N, \quad \text{w.h.p.} \]

\[ \Rightarrow V_N = O(1/\sqrt{N}) \text{ from CLT} \]

\[ \Rightarrow V_N = O(1/N) \text{ for a specific function class [Lee et al '98]} \]
From statistical learning literature we know that

\[ \sup_w |R(w) - R_N(w)| \leq V_N, \quad \text{w.h.p.} \]

\[ \Rightarrow V_N = O(1/\sqrt{N}) \text{ from CLT} \]

\[ \Rightarrow V_N = O(1/N) \text{ for a specific function class [Lee et al '98]} \]

There is no need to minimize \( R_N(w) \) beyond accuracy \( O(V_N) \)
From statistical learning literature we know that
\[ \sup_w |R(w) - R_N(w)| \leq V_N, \quad \text{w.h.p.} \]

\[ \Rightarrow V_N = O(1/\sqrt{N}) \text{ from CLT} \]
\[ \Rightarrow V_N = O(1/N) \text{ for a specific function class [Lee et al '98]} \]

There is no need to minimize \( R_N(w) \) beyond accuracy \( O(V_N) \)

**Goal:** Minimize the risk \( R_N \) up to its statistical accuracy \( V_N \)
\[ R_N(w_N) - R_N(w_N^*) = O(V_N) \]

We say \( w_N \) is within statistical accuracy of \( R_N \) if it satisfies this condition.
Challenges

- Number of observations $N$ is very large

$N \approx 10^8$

$N \approx 10^7$
Challenges

- Number of observations $N$ is very large

$N \approx 10^8$

- Number of data features $d$ is very large

$N \approx 10^7$

- As a result, the problem condition number $\kappa$ is large

- Second-order methods are essential for such ill-conditioned problems!
The implementation of Newton-type methods faces the following challenges:
Shortcomings of Second-order Methods

The implementation of Newton-type methods faces the following challenges:

- Their computational cost per iteration scales with $N$
  - Has been addressed by stochastic and incremental methods

- Their cost scales poorly with the problem dimension $d$
- They require using a line-search scheme for the choice of stepsize
  - Can't always use stepsize $\eta = 1$
- Their fast (quadratic/superlinear) rates appear in a local nbhd. of the solution
  - Fast convergence happens when statistical accuracy is already achieved!
Shortcomings of Second-order Methods

The implementation of Newton-type methods faces the following challenges:

- Their computational cost per iteration scales with $N$
  \[ \Rightarrow \text{Has been addressed by stochastic and incremental methods} \]

- Their cost scales poorly with the problem dimension $d$
  \[ \Rightarrow \text{Has been addressed by sketch ideas and limited-memory schemes} \]
The implementation of Newton-type methods faces the following challenges:

- Their computational cost per iteration scales with $N$
  - Has been addressed by stochastic and incremental methods

- Their cost scales poorly with the problem dimension $d$
  - Has been addressed by sketch ideas and limited-memory schemes

- They require using a line-search scheme for the choice of stepsize
  - Can’t always use stepsize $\eta = 1$
The implementation of Newton-type methods faces the following challenges:

- Their computational cost per iteration scales with $N$
  - Has been addressed by stochastic and incremental methods

- Their cost scales poorly with the problem dimension $d$
  - Has been addressed by sketch ideas and limited-memory schemes

- They require using a line-search scheme for the choice of stepsize
  - Can’t always use stepsize $\eta = 1$

- Their fast (quadratic/superlinear) rates appear in a local nbhd. of the solution
  - Fast convergence happens when statistical accuracy is already achieved!
Observation: All samples are drawn from a common distribution
⇒ Solutions of ERM problems are close if the sample sets have overlaps
Observation: All samples are drawn from a common distribution
⇒ Solutions of ERM problems are close if the sample sets have overlaps

\( w^*_m \) is the solution of \( R_m \)
\( w^*_n \) is the solution of \( R_n \)
If \( n \) samples contain \( m \) samples
⇒ \( w^*_m \) is close to \( w^*_n \)

Figure: ERM with \( m \) and \( n \) samples, \( n > m \)
Adaptive sample size methods

 Observation: All samples are drawn from a common distribution
 ⇒ Solutions of ERM problems are close if the sample sets have overlaps

 ▷ $w^*_m$ is the solution of $R_m$
 ▷ $w^*_n$ is the solution of $R_n$
 ▷ If $n$ samples contain $m$ samples
 ⇒ $w^*_m$ is close to $w^*_n$

 Figure: ERM with $m$ and $n$ samples, $n > m$

Theorem: [Mokhtari et al., NeurIPS, ’16]

Consider $S_m$ and $S_n$ such that $S_m \subseteq S_n \subseteq T$, with $m$ and $n$ samples, respectively. Consider the empirical risks $R_m$ and $R_n$ defined based on $S_m$ and $S_n$, respectively. Assume that $w_m$ solves the ERM problem of $R_m$ within its statistical accuracy, i.e., $R_m(w_m) - R_m(w^*_m) \leq V_m$. Then,

$$R_n(w_m) - R_n(w^*_n) \leq V_m + \frac{2(n - m)}{n} (V_{n-m} + V_m).$$
Adaptive sample size scheme

- Find $w_m$ within the statistical accuracy of $R_m$ with $m$ samples
- Increase sample size to $n > m$ samples ($n = \alpha m$ where $\alpha > 1$)
- Use $w_m$ as a warm start to find approx. solution $w_n$ for $R_n$
Adaptive sample size Newton method (Ada Newton)

- If we properly increase the size of training set:
  - In this picture, $m < n$
  - **Statistical accuracy ball of** $R_m$ **is within Newton quadratic convergence ball of** $R_n$.
  - Then, $w_m$ is within Newton quadratic convergence ball of $R_n$
  - Stepsize is always $\eta = 1$
  - Ada Newton exploits quadratic rate throughout the entire learning process
Adaptive sample size Newton method (Ada Newton)

- If we properly increase the size of training set:
  - In this picture, $m < n$
  - Statistical accuracy ball of $R_m$ is within Newton quadratic convergence ball of $R_n$.
  - Then, $w_m$ is within Newton quadratic convergence ball of $R_n$
  - Stepsize is always $\eta = 1$
  - Ada Newton exploits quadratic rate throughout the entire learning process

Ada Newton
- Step 1: Find $w_m$ that solves $R_m$ to its statistical accuracy $V_m$
- Step 2: Increase the size of training set $n = \alpha m$ ($\alpha > 1$)
- Step 3: Apply single Newton update: $w_n = w_m - \nabla^2 R_n(w_m)^{-1} \nabla R_n(w_m)$
- Step 4: Go back to Step 2 if $n < N$
Doubling the size of training set

**Theorem: (informal) [Mokhtari et al., NeurIPS, ’16]**

*If the size of initial training set $m_0$ is sufficiently large, we can double the size of training set $\alpha = 2$ at each stage, and solve each subproblem with only one step of Newton’s method (with stepsize $\eta = 1$).*
Theorem: (informal) [Mokhtari et al., NeurIPS, ’16]

If the size of initial training set $m_0$ is sufficiently large, we can double the size of training set $\alpha = 2$ at each stage, and solve each subproblem with only one step of Newton’s method (with stepsize $\eta = 1$).

- AdaNewton achieves the statistical accuracy of the full training set
  - After about 2 passes over the data
  - After inversion of about $\log_2 N$ Hessians
Doubling the size of training set

Theorem: (informal) [Mokhtari et al., NeurIPS, ’16]

If the size of initial training set \( m_0 \) is sufficiently large, we can double the size of training set \( \alpha = 2 \) at each stage, and solve each subproblem with only one step of Newton’s method (with stepsize \( \eta = 1 \)).

- AdaNewton achieves the statistical accuracy of the full training set
  - After about 2 passes over the data
  - After inversion of about \( \log_2 N \) Hessians

- Shortcomings:
  - Requires computing \( \log_2 N \) Newton directions (costly for large \( d \))
  - Requires computing \( 2N \) Hessians
Quasi-Newton methods

- Quasi-Newton update for minimizing $F(w)$:

$$w^+ = w - \eta \ H \ \nabla F(w)$$

- where $H$ is close to $\nabla^2 F(w)^{-1}$. 

Local superlinear convergence $\Rightarrow \lim_{t \to \infty} \|w_{t+1} - w^*\| / \|w_t - w^*\| = 0$

This result is asymptotic! 

To use QN updates for adaptive sample size scheme, we need a finite-time analysis.
Quasi-Newton update for minimizing $F(w)$:

$$w^+ = w - \eta \, H \, \nabla F(w)$$

where $H$ is close to $\nabla^2 F(w)^{-1}$.

Variable variation: $s = w^+ - w$, Gradient variation: $y = \nabla F(w^+) - \nabla F(w)$

\[ H^+ = H - \frac{Hyy^T H}{y^T Hy} + \frac{ss^T}{s^T y}, \quad \text{DFP} \]

\[ H^+ = \left( I - \frac{sy^T}{s^T y} \right) H \left( I - \frac{ys^T}{s^T y} \right) + \frac{ss^T}{s^T y}, \quad \text{BFGS} \]
Quasi-Newton methods

- Quasi-Newton update for minimizing $F(w)$:

$$w^+ = w - \eta \ H \ \nabla F(w)$$

- where $H$ is close to $\nabla^2 F(w)^{-1}$.

- Variable variation: $s = w^+ - w$, Gradient variation: $y = \nabla F(w^+ - \nabla F(w)$

$$H^+ = H - \frac{Hyy^T H}{y^T Hy} + \frac{ss^T}{s^T y}, \quad \text{DFP}$$

$$H^+ = \left( I - \frac{sy^T}{s^T y} \right) H \left( I - \frac{ys^T}{s^T y} \right) + \frac{ss^T}{s^T y}, \quad \text{BFGS}$$

- Local superlinear convergence $\Rightarrow \lim_{t \to \infty} \frac{\|w_{t+1} - w^*\|}{\|w_t - w^*\|} = 0$

- This result is asymptotic!

  $\Rightarrow$ To use QN updates for adaptive sample size scheme, we need a finite-time analysis
Recently, we showed that the iterates of QN methods locally converge at a rate of $O((1/\sqrt{t})^t)$, where $t$ is the number of iterations.

Theorem: [Jin & Mokhtari, ’20]

Consider the function $F(w)$ which is $\mu$-strongly convex, $M$-smooth, and its Hessian is Lipschitz continuous with $L$. If the initial point $w_0$ and initial Hessian inverse approximation matrix $H_0$ satisfy

$$\|w_0 - w^*\| \leq \frac{\mu^{\frac{3}{2}}}{200L\sqrt{M}}, \quad \|\nabla^2 F(w^*)^{\frac{1}{2}}H_0\nabla^2 F(w^*)^{\frac{1}{2}} - I\|_F \leq \frac{1}{12},$$

then, the iterates of DFP/BFGS converge to $w^*$ at a superlinear rate of

$$\|w_t - w^*\| \leq \sqrt{\frac{L}{\mu}} \left( \frac{1}{\sqrt{t}} \right)^t \|w_0 - w^*\|.$$
Adaptive sample size quasi-Newton method (Ada QN)

- Statistical accuracy ball of $R_m$ is within QN superlinear convergence ball of $R_n$
- $w_m$ is within QN superlinear convergence ball of $R_n$
- A few quasi-Newton updates yield $w_n$ within statistical accuracy of $R_n$
- Stepsize is always $\eta = 1$
- AdaQN exploits superlinear rate of QN algs throughout the entire learning process
Adaptive sample size quasi-Newton method (Ada QN)

- Statistical accuracy ball of $R_m$ is within QN superlinear convergence ball of $R_n$
- $w_m$ is within QN superlinear convergence ball of $R_n$
- A few quasi-Newton updates yield $w_n$ within statistical accuracy of $R_n$
- Stepsize is always $\eta = 1$
- AdaQN exploits superlinear rate of QN algs throughout the entire learning process

- If we double the size of training set at each phase, are the local convergence neighborhood conditions satisfied?

\[
\|w_m - w_n^*\| \leq \frac{\mu^{3/2}}{200L\sqrt{M}}, \quad \|\nabla^2 R_n(w_n^*)^{-\frac{1}{2}}H_0\nabla^2 R_n(w_n^*)^{-\frac{1}{2}} - I\|_F \leq \frac{1}{12}
\]

- How should we come up with a computationally-efficient $H_0$ for $R_n$?
Adaptive sample size quasi-Newton method (Ada QN)

- Use the solution of last problem as the initial point
- Set the initial Hessian inverse approximation as $H_0 = \nabla^2 R_{m_0}(w_{m_0})^{-1}$
- Then, the required conditions for local superlinear rate are satisfied!
Adaptive sample size quasi-Newton method (Ada QN)

- Use the solution of last problem as the initial point
- Set the initial Hessian inverse approximation as $H_0 = \nabla^2 R_{m_0}(w_{m_0})^{-1}$
- Then, the required conditions for local superlinear rate are satisfied!
Main Results

**Theorem: (informal) [Jin & Mokhtari, ’21]**

*If the size of initial training set $m_0$ is sufficiently large, we can double the size of training set $\alpha = 2$ at each stage, and solve each subproblem after at most three steps of DFP/BFGS (with stepsize $\eta = 1$).*

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>gradient comp.</th>
<th>Hessian comp.</th>
<th>matrix-vec product</th>
<th>matrix inversion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ada Newton</td>
<td>$O(N)$</td>
<td>$O(N)$</td>
<td>$O(\log N)$</td>
<td>$O(\log N)$</td>
</tr>
<tr>
<td>Ada QN</td>
<td>$O(N)$</td>
<td>$m_0$</td>
<td>$O(\log N)$</td>
<td>1</td>
</tr>
</tbody>
</table>

**Table:** Overall complexity of Ada Newton and Ada QN to reach the statistical accuracy of the full training set with $N$ samples. $O(.)$ notation only hides absolute constants.
Numerical results

- Logistic regression problem with the Epsilon dataset $N = 65,000, d = 2000$
- $m_0 = 1000$, best performance $\mu = 10^{-4} \Rightarrow$ condition number $\kappa \approx 10^4$

![Graphs showing training error and test error vs. runtime](image)

**Figure:** Training error (left) and Test error (right) vs. runtime
Discussed challenges in using second-order methods for large-scale ERM

- Adaptive sample size second-order methods
  - Unit stepsize
  - No line search
  - Not sensitive to initial point
  - Less Hessian inversion
  - Exploit quadratic/superlinear convergence during entire training process

Extension to nonconvex settings is also possible! [Mokhtari et al., AISTATS, '19]

For strongly Morse functions

Moving between second-order stationary points of ERM problems
Summary

- Discussed challenges in using second-order methods for large-scale ERM

- Adaptive sample size second-order methods
  - Unit stepsize
  - No line search
  - Not sensitive to initial point
  - Less Hessian inversion
  - Exploit quadratic/superlinear convergence during entire training process

Extension to nonconvex settings is also possible! [Mokhtari et al., AISTATS, '19]

For strongly Morse functions

Moving between second-order stationary points of ERM problems
Summary

- Discussed challenges in using second-order methods for large-scale ERM

- Adaptive sample size second-order methods
  - Unit stepsize
  - No line search
  - Not sensitive to initial point
  - Less Hessian inversion
  - Exploit quadratic/superlinear convergence during entire training process

- Extension to nonconvex settings is also possible! [Mokhtari et al., AISTATS, '19]
  - For strongly Morse functions
  - Moving between second-order stationary points of ERM problems
Thanks for your attention!


