SECOND ORDER MACHINE LEARNING

Michael W. Mahoney

ICSI and Department of Statistics
UC Berkeley
Machine Learning’s “Inverse” Problem

Your choice:

- 1st Order Methods: FLAG n’ FLARE, or
  - disentangle geometry from sequence of iterates

- 2nd Order Methods: Stochastic Newton-Type Methods
  - “simple” methods for convex
  - “more subtle” methods for non-convex
Big Data ... Massive Data ...

Sequencers begin giving flurries of data

Doubling Time = 6.76 ± 0.33

Simultaneously Recorded Neurons

Publication Date

Experiments Production Data in CRISTK
Humongous Data ...

We've decided to take big data to the next level...

© D. Fletcher for CloudTweaks.com
How do we view BIG data?
Computer Scientists

- Data: are a record of everything that happened.
- Goal: process the data to find interesting patterns and associations.
- Methodology: Develop approximation algorithms under different models of data access since the goal is typically computationally hard.

Statisticians (and Natural Scientists, etc)

- Data: are a particular random instantiation of an underlying process describing unobserved patterns in the world.
- Goal: is to extract information about the world from noisy data.
- Methodology: Make inferences (perhaps about unseen events) by positing a model that describes the random variability of the data around the deterministic model.
... ARE VERY DIFFERENT PARADIGMS

Statistics, natural sciences, scientific computing, etc:

- Problems often involve computation, but the study of computation per se is secondary
- Only makes sense to develop algorithms for well-posed problems \(^1\)
- First, write down a model, and think about computation later

Computer science:

- Easier to study computation per se in discrete settings, e.g., Turing machines, logic, complexity classes
- Theory of algorithms divorces computation from data
- First, run a fast algorithm, and ask what it means later

\(^1\) Solution exists, is unique, and varies continuously with input data
**Problem Statement**

**Problem 1: Composite Optimization Problem**

\[
\min_{x \in \mathcal{X} \subseteq \mathbb{R}^d} F(x) = f(x) + h(x)
\]

- \(f\): Convex and Smooth
- \(h\): Convex and (Non-)Smooth

**Problem 2: Minimizing Finite Sum Problem**

\[
\min_{x \in \mathcal{X} \subseteq \mathbb{R}^d} F(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x)
\]

- \(f_i\): (Non-)Convex and Smooth
- \(n \gg 1\)
Efficient and Effective Optimization Methods

Modern “Big-Data”

- Classical Optimization Algorithms
  - Effective but Inefficient

Need to design variants, that are:

1. **Efficient**, i.e., Low Per-Iteration Cost
2. **Effective**, i.e., Fast Convergence Rate
Scientific Computing and Machine Learning share the same challenges, and use the same means, but to get to different ends!

Machine Learning has been, and continues to be, very busy designing efficient and effective optimization methods.
First Order Methods

- Variants of Gradient Descent (GD):
  - Reduce the per-iteration cost of GD ⇒ Efficiency
  - Achieve the convergence rate of the GD ⇒ Effectiveness

\[ x^{(k+1)} = x^{(k)} - \alpha_k \nabla F(x^{(k)}) \]
First Order Methods

- E.g.: SAG, SDCA, SVRG, Prox-SVRG, Acc-Prox-SVRG, Acc-Prox-SDCA, S2GD, mS2GD, MISO, SAGA, AMSVRG, ...
But why?

Q: Why do we use (stochastic) 1st order method?

- **Cheaper Iterations?** i.e., $n \gg 1$ and/or $d \gg 1$

- **Avoids Over-fitting?**
Challenges with “simple” 1st order method for “over-fitting”:

- Highly sensitive to ill-conditioning
- Very difficult to tune (many) hyper-parameters

“Over-fitting” is difficult with “simple” 1st order method!
Remedy?

1. "Not-So-Simple" 1st order method, e.g., accelerated and adaptive

2. 2nd order methods, e.g.,

\[
x^{(k+1)} = x^{(k)} - [\nabla^2 F(x^{(k)})]^{-1} \nabla F(x^{(k)})
\]
Your Choice Of....
“Not-So-Simple” 1st order method: FLAG n’ FLARE

1. **Problem 1: Composite Optimization Problem**

\[
\min_{x \in \mathcal{X} \subseteq \mathbb{R}^d} F(x) = f(x) + h(x)
\]

- \(f\): Convex and Smooth, \(h\): Convex and (Non-)Smooth

2. **Problem 2: Minimizing Finite Sum Problem**

\[
\min_{x \in \mathcal{X} \subseteq \mathbb{R}^d} F(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x)
\]

- \(f_i\): (Non-)Convex and Smooth, \(n \gg 1\)

2nd order methods: Stochastic Newton-Type Methods
- Stochastic Newton, Trust Region, Cubic Regularization
Efficient and Effective Optimization Methods

COLLABORATORS

• FLAG n’ FLARE
  • Fred Roosta (UC Berkeley)
  • Xiang Cheng (UC Berkeley)
  • Stefan Palombo (UC Berkeley)
  • Peter L. Bartlett (UC Berkeley & QUT)

• Sub-Sampled Newton-Type Methods for Convex
  • Fred Roosta (UC Berkeley)
  • Peng Xu (Stanford)
  • Jiyan Yang (Stanford)
  • Christopher Ré (Stanford)

• Sub-Sampled Newton-Type Methods for Non-convex
  • Fred Roosta (UC Berkeley)
  • Peng Xu (Stanford)

• Implementations on GPU, etc.
  • Fred Roosta (UC Berkeley)
  • Sudhir Kylasa (Purdue)
  • Ananth Grama (Purdue)
Subgradient Method

Composite Optimization Problem

\[
\min_{\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^d} F(\mathbf{x}) = f(\mathbf{x}) + h(\mathbf{x})
\]

- \( f \): Convex (Non-)Smooth
- \( h \): Convex (Non-)Smooth
Subgradient Method

Algorithm 1 Subgradient Method

1: **Input**: $x_1$, and $T$
2: **for** $k = 1, 2, \ldots, T - 1$ **do**
3: \hspace{1em} $g_k \in \partial (f(x_k) + h(x_k))$
4: \hspace{1em} $x_{k+1} = \arg \min_{x \in \mathcal{X}} \left\{ \langle g_k, x \rangle + \frac{1}{2\alpha_k} \| x - x_k \|^2 \right\}$
5: **end for**
6: **Output**: $\bar{x} = \frac{1}{T} \sum_{t=1}^{T} x_t$

- $\alpha_k$: Step-size
  - Constant Step-size: $\alpha_k = \alpha$
  - Diminishing Step size $\sum_{k=1}^{\infty} \alpha_k = \infty$, $\lim_{k \to \infty} \alpha_k = 0$
Example: Logistic Regression

- \{a_i, b_i\}: features and labels
- \(a_i \in \{0, 1\}^d, \; b_i \in \{0, 1\}\)

\[
F(x) = \sum_{i=1}^{n} \log(1 + e^{\langle a_i, x \rangle}) - b_i \langle a_i, x \rangle
\]

\[
\nabla F(x) = \sum_{i=1}^{n} \left( \frac{1}{1 + e^{-\langle a_i, x \rangle}} - b_i \right) a_i
\]

Infrequent Features \(\Rightarrow\) Small Partial Derivative
PREDICTIVE VS. IRRELEVANT FEATURES

- Very infrequent features $\Rightarrow$ Highly predictive (e.g. “CANON” in document classification)

- Very frequent features $\Rightarrow$ Highly irrelevant (e.g. “and” in document classification)
AdaGrad [Duchi et al., 2011]

- **Frequent** Features $\Rightarrow$ **Large** Partial Derivative $\Rightarrow$ Learning Rate $\downarrow$

- **Infrequent** Features $\Rightarrow$ **Small** Partial Derivative $\Rightarrow$ Learning Rate $\uparrow$

Replace $\alpha_k$ with **scaling matrix** adaptively...

Many follows up works: RMSProp, Adam, Adadelta, etc...
AdaGrad [Duchi et al., 2011]

Algorithm 2 AdaGrad

1: **Input:** $x_1$, $\eta$ and $T$
2: **for** $k = 1, 2, \ldots, T - 1$ **do**
3:   - $g_k \in \partial f(x_k)$
4:   - Form scaling matrix $S_k$ based on $\{g_t; t = 1, \ldots, k\}$
5:   - $x_{k+1} = \arg\min_{x \in \mathcal{X}} \left\{ \langle g_k, x \rangle + h(x) + \frac{1}{2} (x - x_k)^T S_k (x - x_k) \right\}$
6: **end for**
7: **Output:** $\bar{x} = \frac{1}{T} \sum_{t=1}^{T} x_t$
Let $x^*$ be an optimum point. We have:

- **AdaGrad [Duchi et al., 2011]**:

$$F(\bar{x}) - F(x^*) \leq O \left( \frac{\sqrt{d} D_\infty \alpha}{\sqrt{T}} \right),$$

where $\alpha \in \left[ \frac{1}{\sqrt{d}}, 1 \right]$ and $D_\infty = \max_{x, y \in \mathcal{X}} \|y - x\|_\infty$, and

- **Subgradient Descent**:

$$F(\bar{x}) - F(x^*) \leq O \left( \frac{D_2}{\sqrt{T}} \right),$$

where $D_2 = \max_{x, y \in \mathcal{X}} \|y - x\|_2.$
**Comparison**

Competitive Factor:

\[
\frac{\sqrt{d} D_\infty \alpha}{D_2}
\]

- $D_\infty$ and $D_2$ depend on geometry of $\mathcal{X}$
- e.g., $\mathcal{X} = \{x; \|x\|_\infty \leq 1\}$ then $D_2 = \sqrt{d} D_\infty$

\[
\alpha = \frac{\sum_{i=1}^d \sqrt{\sum_{t=1}^T (g_t)_i^2}}{\sqrt{d \sum_{t=1}^T \|g_t\|^2}}
\]

depends on $\{g_t; t = 1, \ldots, T\}$

**Problem Statement**

**Problem 1: Composite Optimization Problem**

\[
\min_{x \in \mathcal{X} \subseteq \mathbb{R}^d} F(x) = f(x) + h(x)
\]

- **f**: Convex and Smooth (w. L-Lipschitz Gradient)
- **h**: Convex and (Non-)Smooth

- Subgradient Methods: \( O \left( \frac{1}{\sqrt{T}} \right) \)
- ISTA: \( O \left( \frac{1}{T} \right) \)
- FISTA [Beck and Teboulle, 2009]: \( O \left( \frac{1}{T^2} \right) \)
Best of both worlds?

- **Accelerated** Gradient Methods \(\Rightarrow\) **Optimal Rate**
  - e.g., \(\frac{1}{T^2}\) vs. \(\frac{1}{T}\) vs. \(\frac{1}{\sqrt{T}}\)

- **Adaptive** Gradient Methods \(\Rightarrow\) **Better Constant**
  - \(\sqrt{d}D_\infty \alpha\) vs. \(D_2\)

*How about Accelerated and Adaptive Gradient Methods?*
- **FLAG**: Fast Linearly-Coupled Adaptive Gradient Method
- **FLARE**: FLAg RElaxed
Algorithm 3 FLAG

1: **Input:** $x_0 = y_0 = z_0$ and $L$
2: **for** $k = 1, 2, \ldots, T$ **do**
3:     - $y_{k+1} = \text{Prox}(x_k)$
4:     - Gradient Mapping $g_k = -L(y_{k+1} - x_k)$
5:     - Form $S_k$ based on $\{\frac{g_t}{\|g_t\|}; t = 1, \ldots, k\}$
6:     - Compute $\eta_k$
7:     - $z_{k+1} = \arg\min_{z \in \mathcal{X}} \langle \eta_k g_k, z - z_k \rangle + \frac{1}{2}(z - z_k)^T S_k (z - z_k)$
8:     - $x_k = \text{Linearly Couple} (y_{k+1}, z_{k+1})$
9: **end for**
10: **Output:** $y_{T+1}$

\[
\text{Prox}(x_k) := \arg\min_{x \in \mathcal{X}} \left\{ \langle \nabla f(x_k), x \rangle + h(x) + \frac{L}{2} \|x - x_k\|_2^2 \right\}
\]
Algorithm 4 Birds Eye View of FLAG

1: **Input**: $x_0$
2: **for** $k = 1, 2, \ldots, T$ **do**
3: \hspace{1em} - $y_k$ : Usual Gradient Step
4: \hspace{1em} - Form Gradient History
5: \hspace{1em} - $z_k$ : Scaled Gradient Step
6: \hspace{1em} - Find mixing weight $w$ via Binary Search
7: \hspace{1em} - $x_{k+1} = (1 - w)y_{k+1} + wz_{k+1}$
8: **end for**
9: **Output**: $y_{T+1}$
Let $\mathbf{x}^*$ be an optimum point. We have:

- **FLAG** [CRPBM, 2016]:

$$F(\bar{\mathbf{x}}) - F(\mathbf{x}^*) \leq O\left(\frac{dD_\infty^2 \beta}{T^2}\right),$$

where $\beta \in [\frac{1}{d}, 1]$ and $D_\infty = \max_{\mathbf{x}, \mathbf{y} \in \mathcal{X}} \|\mathbf{y} - \mathbf{x}\|_\infty$, and

- **FISTA** [Beck and Teboulle, 2009]:

$$F(\bar{\mathbf{x}}) - F(\mathbf{x}^*) \leq O\left(\frac{D_2^2}{T^2}\right),$$

where $D_2 = \max_{\mathbf{x}, \mathbf{y} \in \mathcal{X}} \|\mathbf{y} - \mathbf{x}\|_2$. 
COMPARIISON

Competitive Factor:

\[
\frac{dD^2_\infty \beta}{D^2_2}
\]

- \(D_\infty\) and \(D_2\) depend on geometry of \(\mathcal{X}\)
  - e.g., \(\mathcal{X} = \{x; \|x\|_\infty \leq 1\}\) then \(D_2 = \sqrt{dD_\infty}\)

\[
\beta = \left( \frac{\sum_{i=1}^{d} \sqrt{\sum_{t=1}^{T} [\tilde{g}_t]_i^2}}{dT} \right)^2
\]

- \(\beta\) depends on \(\{\tilde{g}_t := g_t / \|g_t\|; t = 1, \ldots, T\}\)
**Linear Coupling**

- Linearly Couple of \((y_{k+1}, z_{k+1})\) via a “\(\epsilon\)-Binary Search”:
- Find \(\epsilon\) approximation to the root of non-linear equation

\[
\langle \text{Prox}(ty + (1 - t)z) - (ty + (1 - t)z), y - z \rangle = 0,
\]

where

\[
\text{Prox}(x) := \arg\min_{y \in \mathcal{C}} h(y) + \frac{L}{2} \|y - (x - \frac{1}{L} \nabla f(x))\|_2^2.
\]

- At most \(\log(1/\epsilon)\) steps using bisection
- At most \(2 + \log(1/\epsilon)\) \(\text{Prox}\) evals per-iteration more than FISTA

Can be Expensive!
**Linear Coupling**

- Linearly approximate:

\[
\langle t \text{Prox}(y) + (1 - t) \text{Prox}(z) - (ty + (1 - t)z), y - z \rangle = 0.
\]

- Linear equation in \( t \), so closed form solution!

\[
t = \frac{\langle z - \text{Prox}(z), y - z \rangle}{\langle (z - \text{Prox}(z)) - (y - \text{Prox}(y)), y - z \rangle}
\]

- At most 2 \( \text{Prox} \) evals per-iteration more than FISTA

- Equivalent to \( \epsilon \)-Binary Search with \( \epsilon = 1/3 \)

Better But Might Not Be Good Enough!
Basic Idea: Choose mixing weight by intelligent “futuristic” guess

- Guess now, and next iteration, correct if guessed wrong

FLARE: exactly the same Prox evals per-iteration as FISTA!

FLARE: has the similar theoretical guarantee as FLAG!
\[ \mathcal{L}(x_1, x_2, \ldots, x_C) = \sum_{i=1}^{n} \sum_{c=1}^{C} -\mathbf{1}(b_i = c) \log \left( \frac{e^{\langle a_i, x_c \rangle}}{1 + \sum_{b=1}^{C-1} e^{\langle a_i, x_b \rangle}} \right) \\
= \sum_{i=1}^{n} \left( \log \left( 1 + \sum_{c=1}^{C-1} e^{\langle a_i, x_c \rangle} \right) - \sum_{c=1}^{C-1} \mathbf{1}(b_i = c) \langle a_i, x_c \rangle \right) \]
Classification: 20 Newsgroups

Prediction across 20 different newsgroups

<table>
<thead>
<tr>
<th>Data</th>
<th>Train Size</th>
<th>Test Size</th>
<th>(d)</th>
<th>Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>20 Newsgroups</td>
<td>10,142</td>
<td>1,127</td>
<td>53,975</td>
<td>20</td>
</tr>
</tbody>
</table>

\[
\min_{\|x\|_\infty \leq 1} \mathcal{L}(x_1, x_2, \ldots, x_C)
\]
Classification: 20 Newsgroups

Log of Objective Function: 20News

Test Accuracy: 20News

Michael W. Mahoney (UC Berkeley)
Predicting forest cover type from cartographic variables

<table>
<thead>
<tr>
<th>DATA</th>
<th>TRAIN SIZE</th>
<th>TEST SIZE</th>
<th>$d$</th>
<th>CLASSES</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoveType</td>
<td>435,759</td>
<td>145,253</td>
<td>54</td>
<td>7</td>
</tr>
</tbody>
</table>

$$
\min_{x \in \mathbb{R}^d} \mathcal{L}(x_1, x_2, \ldots, x_C) + \lambda \|x\|_1
$$
Classification: Forest CoverType

Log of Objective Function: covtype

Test Accuracy: covtype
Regression: BlogFeedback

Prediction of the number of comments in the next 24 hours for blogs

<table>
<thead>
<tr>
<th>DATA</th>
<th>TRAIN SIZE</th>
<th>TEST SIZE</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>BlogFeedback</td>
<td>47,157</td>
<td>5,240</td>
<td>280</td>
</tr>
</tbody>
</table>

\[
\min_{x \in \mathbb{R}^d} \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1
\]
Regression: BlogFeedback

Log of Objective Function: blog-feedback

Test Error: blog-feedback

Michael W. Mahoney (UC Berkeley)  Second order machine learning
2nd order methods: Stochastic Newton-Type Methods

- Stochastic Newton (think: convex)
- Stochastic Trust Region (think: non-convex)
- Stochastic Cubic Regularization (think: non-convex)

Problem 2: Minimizing Finite Sum Problem

\[
\min_{x \in \mathcal{X} \subseteq \mathbb{R}^d} F(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x)
\]

- \( f_i \): (Non-)Convex and Smooth
- \( n \gg 1 \)
Second Order Methods

- Use both gradient and Hessian information
- Fast convergence rate
- Resilient to ill-conditioning
- They “over-fit” nicely!
- However, per-iteration cost is high!
Sensorless Drive Diagnosis

\[ n : 50,000, \ p = 528, \ \text{No. Classes} = 11, \ \lambda : 0.0001 \]

**Figure:** Test Accuracy
Sensorless Drive Diagnosis

\[ n : 50,000, \; p = 528, \; \text{No. Classes} = 11, \; \lambda : 0.0001 \]

**Figure:** Time/Iteration

- Adagrad: reg:1.00e-04, step:1.00e-02
- RMSProp: reg:1.00e-04, step:1.00e-02
- GD: reg:1.00e-04, step:1.00e-03
- Adam: reg:1.00e-04, step:1.00e-02
- Adadelta: reg:1.00e-04, step:1.00e-02
- Newton: reg:1.00e-04
Second-order methods: Stochastic Newton-Type Methods

**SECOND ORDER METHODS**

- **Deterministically approximating second order information cheaply**
  - Quasi-Newton, e.g., BFGS and L-BFGS [Nocedal, 1980]

- **Randomly approximating second order information cheaply**
  - Sketching the Hessian [Pilanci et al., 2015]
  - Sub-Sampling the Hessian and the gradient [RM-I & RM-II, 2016, Bollapragada et al., 2016, ...]
Iterative Scheme

\[ x^{(k+1)} = \arg \min_{x \in \mathcal{D} \cap \mathcal{X}} \left\{ F(x^{(k)}) + (x - x^{(k)})^T g(x^{(k)}) + \frac{1}{2\alpha_k} (x - x^{(k)})^T H(x^{(k)})(x - x^{(k)}) \right\} \]
Second-order methods: Stochastic Newton-Type Methods

**HESSIAN SUB-SAMPING**

\[ g(x) = \nabla F(x) \]

\[ H(x) = \frac{1}{|S|} \sum_{j \in S} \nabla^2 f_j(x) \]
First, let’s consider the convex case....
Convex Problems

- Each $f_i$ is smooth and \textit{weakly} convex
- $F$ is $\gamma$-\textit{strongly} convex
“We want to design methods for machine learning that are not as ideal as Newton’s method but have [these] properties: first of all, they tend to turn towards the right directions and they have the right length, [i.e.,] the step size of one is going to be working most of the time...and we have to have an algorithm that scales up for machine leaning.”

Prof. Jorge Nocedal

IPAM Summer School, 2012
Tutorial on Optimization Methods for ML
(Video - Part I: 50’ 03” )
What do we need?

- Requirements:
  
  (R.1) **Scale up:**

  (R.2) **Turn to right directions:**

  (R.3) **Not ideal but close:**

  (R.4) **Right step length:**
What do we need?

- Requirements:

  (R.1) **Scale up:** $|S|$ must be independent of $n$, or at least smaller than $n$ and for $p \gg 1$, allow for inexactness

  (R.2) **Turn to right directions:**

  (R.3) **Not ideal but close:**

  (R.4) **Right step length:**
What do we need?

- **Requirements:**

  (R.1) **Scale up:** $|S|$ must be independent of $n$, or at least smaller than $n$ and for $p \gg 1$, allow for inexactness

  (R.2) **Turn to right directions:** $H(x)$ must preserve the spectrum of $\nabla^2 F(x)$ as much as possible

  (R.3) **Not ideal but close:**

  (R.4) **Right step length:**
What do we need?

- Requirements:

  (R.1) **Scale up**: \(|S|\) must be independent of \(n\), or at least smaller than \(n\) and for \(p \gg 1\), allow for inexactness

  (R.2) **Turn to right directions**: \(H(x)\) must preserve the spectrum of \(\nabla^2 F(x)\) as much as possible

  (R.3) **Not ideal but close**: Fast local convergence rate, close to that of Newton

  (R.4) **Right step length**:
Requirements:

(R.1) **Scale up:** $|S|$ must be independent of $n$, or at least smaller than $n$ and for $p \gg 1$, allow for inexactness

(R.2) **Turn to right directions:** $H(x)$ must preserve the spectrum of $\nabla^2 F(x)$ as much as possible

(R.3) **Not ideal but close:** Fast local convergence rate, close to that of Newton

(R.4) **Right step length:** Unit step length eventually works
Requirements:

(R.1) **Scale up:** $|S|$ must be independent of $n$, or at least smaller than $n$ and for $p \gg 1$, allow for inexactness.

(R.2) **Turn to right directions:** $H(x)$ must preserve the spectrum of $\nabla^2 F(x)$ as much as possible.

(R.3) **Not ideal but close:** Fast local convergence rate, close to that of Newton.

(R.4) **Right step length:** Unit step length eventually works.
**Lemma (Uniform Hessian Sub-Sampling)**

Given any $0 < \epsilon < 1$, $0 < \delta < 1$ and $\mathbf{x} \in \mathbb{R}^p$, if

$$\left| S \right| \geq \frac{2 \kappa^2 \ln(2p/\delta)}{\epsilon^2},$$

then

$$\text{Pr} \left( (1 - \epsilon) \nabla^2 F(\mathbf{x}) \preceq H(\mathbf{x}) \preceq (1 + \epsilon) \nabla^2 F(\mathbf{x}) \right) \geq 1 - \delta.$$
Requirements:

(R.1) **Scale up:** $|S|$ must be independent of $n$, or at least smaller than $n$ and for $p \gg 1$, allow for inexactness

(R.2) **Turn to right directions:** $H(x)$ must preserve the spectrum of $\nabla^2 F(x)$ as much as possible

(R.3) **Not ideal but close:** Fast local convergence rate, close to that of Newton

(R.4) **Right step length:** Unit step length eventually works
**Theorem (Error Recursion)**

Using $\alpha_k = 1$, with high-probability, we have

$$
\|x^{(k+1)} - x^*\| \leq \rho_0 \|x^{(k)} - x^*\| + \xi \|x^{(k)} - x^*\|^2,
$$

where

$$
\rho_0 = \frac{\epsilon}{(1 - \epsilon)}, \quad \text{and} \quad \xi = \frac{L}{2(1 - \epsilon)\gamma}.
$$

- $\rho_0$ is problem-independent! $\Rightarrow$ Can be made arbitrarily small!
SSN-H: \textbf{Q-Linear Convergence}

**Theorem (Q-Linear Convergence)**

Consider any $0 < \rho_0 < \rho < 1$ and $\epsilon \leq \rho_0 / (1 + \rho_0)$. If

$$
\|x^{(0)} - x^*\| \leq \frac{\rho - \rho_0}{\xi},
$$

we get locally \textit{Q-linear} convergence

$$
\|x^{(k)} - x^*\| \leq \rho \|x^{(k-1)} - x^*\|, \quad k = 1, \ldots, k_0
$$

with high-probability.

Possible to get \textit{superlinear} rate as well.
Requirements:

(R.1) **Scale up**: $|S|$ must be independent of $n$, or at least smaller than $n$ and for $p \gg 1$, allow for inexactness

(R.2) **Turn to right directions**: $H(x)$ must preserve the spectrum of $\nabla^2 F(x)$ as much as possible

(R.3) **Not ideal but close**: Fast local convergence rate, close to that of Newton

(R.4) **Right step length**: Unit step length eventually works
**Lemma (Uniform Hessian Sub-Sampling)**

Given any $0 < \epsilon < 1$, $0 < \delta < 1$, and $\mathbf{x} \in \mathbb{R}^p$, if

$$|S| \geq \frac{2\kappa \ln(p/\delta)}{\epsilon^2},$$

then

$$\Pr \left( (1 - \epsilon) \gamma \leq \lambda_{\min} (H(\mathbf{x})) \right) \geq 1 - \delta.$$
SSN-H: **Inexact Update**

Assume $\mathcal{X} = \mathbb{R}^p$

Descent Dir.:  
\[
\{ \| H(x^{(k)}) p_k + \nabla F(x^{(k)}) \| \leq \theta_1 \| \nabla F(x^{(k)}) \| \}
\]

Step Size:  
\[
\alpha_k = \arg \max_\alpha \quad \text{s.t. } \alpha \leq 1 \\
F(x^{(k)} + \alpha p_k) \leq F(x^{(k)}) + \alpha \beta p_k^T \nabla F(x^{(k)})
\]

Update:  
\[
x^{(k+1)} = x^{(k)} + \alpha_k p_k
\]

$0 < \beta, \theta_1, \theta_2 < 1$
SSN-H Algorithm: Inexact Update

Algorithm 5 Globally Convergent SSN-H with inexact solve

1: Input: $x^{(0)}$, $0 < \delta < 1$, $0 < \epsilon < 1$, $0 < \beta, \theta_1, \theta_2 < 1$
2: - Set the sample size, $|S|$, with $\epsilon$ and $\delta$
3: for $k = 0, 1, 2, \cdots$ until termination do
4:   - Select a sample set, $S$, of size $|S|$ and form $H(x^{(k)})$
5:   - Update $x^{(k+1)}$ with $H(x^{(k)})$ and inexact solve
6: end for
Theorem (Global Convergence of Algorithm 5)

Using Algorithm 5 with $\theta_1 \approx 1/\sqrt{\kappa}$, with high-probability, we have

$$F(x^{(k+1)}) - F(x^*) \leq (1 - \rho) (F(x^{(k)}) - F(x^*)),$$

where $\rho = \alpha_k \beta / \kappa$ and $\alpha_k \geq \frac{2(1 - \theta_2)(1 - \beta)(1 - \epsilon)}{\kappa}$. 


Local + Global

**Theorem**

For any $\rho < 1$ and $\epsilon \approx \rho / \sqrt{\kappa}$, Algorithm 5 is globally convergent and after $O(\kappa^2)$ iterations, with high-probability achieves “problem-independent” Q-linear convergence, i.e.,

$$\|x^{(k+1)} - x^*\| \leq \rho \|x^{(k)} - x^*\|.$$

Moreover, the step size of $\alpha_k = 1$ passes Armijo rule for all subsequent iterations.
“Any optimization algorithm for which the unit step length works has some wisdom. It is too much of a fluke if the unit step length [accidentally] works.”

Prof. Jorge Nocedal
IPAM Summer School, 2012
Tutorial on Optimization Methods for ML
(Video - Part I: 56' 32'')
So far these efforts mostly treated convex problems....

Now, it is time for non-convexity!
Non-Convex Is Hard!

- Saddle points, Local Minima, Local Maxima
- Optimization of a degree four polynomial: NP-hard [Hillar et al., 2013]
- Checking whether a point is not a local minimum: NP-complete [Murty et al., 1987]
All convex problems are the same, while every non-convex problem is different.

Not sure who’s quote this is!
(ε_g, ε_H) — Optimality

\[ \| \nabla F(x) \| \leq \epsilon_g, \]

\[ \lambda_{\text{min}}(\nabla^2 F(x)) \geq -\epsilon_H \]

\[
\begin{align*}
\mathbf{s}^{(k)} &= \arg\min_{\|\mathbf{s}\| \leq \Delta_k} \langle \mathbf{s}, \nabla F(\mathbf{x}^{(k)}) \rangle + \frac{1}{2} \langle \mathbf{s}, \nabla^2 F(\mathbf{x}^{(k)}) \mathbf{s} \rangle \\
\end{align*}
\]


\[
\begin{align*}
\mathbf{s}^{(k)} &= \arg\min_{\mathbf{s} \in \mathbb{R}^d} \langle \mathbf{s}, \nabla F(\mathbf{x}^{(k)}) \rangle + \frac{1}{2} \langle \mathbf{s}, \nabla^2 F(\mathbf{x}^{(k)}) \mathbf{s} \rangle + \frac{\sigma_k}{3} \|\mathbf{s}\|^3
\end{align*}
\]
To get iteration complexity, all previous work required:

$$\left\| \left( H(x^{(k)}) - \nabla^2 F(x^{(k)}) \right) s^{(k)} \right\| \leq C \|s^{(k)}\|^2$$  \hspace{1cm} (1)

Stronger than “Dennis-Moré”

$$\lim_{k \to \infty} \frac{\left\| \left( H(x(k)) - \nabla^2 F(x(k)) \right) s(k) \right\|}{\|s(k)\|} = 0$$

We relaxed (1) to

$$\left\| \left( H(x^{(k)}) - \nabla^2 F(x^{(k)}) \right) s^{(k)} \right\| \leq \epsilon \|s^{(k)}\|$$  \hspace{1cm} (2)

Quasi-Newton, Sketching, Sub-Sampling satisfy Dennis-Moré and (2) but not necessarily (1)
Recall...

\[ F(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x) \]
**Lemma (Complexity of Uniform Sampling)**

Suppose $\|\nabla^2 f_i(x)\| \leq K$, $\forall i$. Given any $0 < \epsilon < 1$, $0 < \delta < 1$, and $x \in \mathbb{R}^d$, if

$$|S| \geq \frac{16K^2}{\epsilon^2} \log \frac{2d}{\delta},$$

then for $H(x) = \frac{1}{|S|} \sum_{j \in S} \nabla^2 f_j(x)$, we have

$$\Pr \left( \|H(x) - \nabla^2 F(x)\| \leq \epsilon \right) \geq 1 - \delta.$$

- Only top eigenvalues/eigenvectors need to be preserved.
Second-order methods: Stochastic Newton-Type Methods

\[ F(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(a_i^T x) \]

\[ p_i = \frac{|f''_i(a_i^T x)| \|a_i\|^2}{\sum_{j=1}^{n} |f''_j(a_j^T x)| \|a_j\|^2} \]
Lemma (Complexity of Non-Uniform Sampling)

Suppose $\|\nabla^2 f_i(x)\| \leq K_i$, $i = 1, 2, \ldots, n$. Given any $0 < \epsilon < 1$, $0 < \delta < 1$, and $x \in \mathbb{R}^d$, if

$$|S| \geq \frac{16 \bar{K}^2}{\epsilon^2} \log \frac{2d}{\delta},$$

then for $H(x) = \frac{1}{|S|} \sum_{j \in S} \frac{1}{n p_j} \nabla^2 f_j(x)$, we have

$$\Pr \left( \|H(x) - \nabla^2 F(x)\| \leq \epsilon \right) \geq 1 - \delta,$$

where

$$\bar{K} = \frac{1}{n} \sum_{i=1}^{n} K_i.$$
Algorithm 6 Stochastic Trust-Region Algorithm

1: Input: $x_0$, $\Delta_0 > 0$ $\eta \in (0, 1)$, $\gamma > 1$, $0 < \epsilon, \epsilon_g, \epsilon_H < 1$

2: for $k = 0, 1, 2, \cdots$ until termination do

3: $s_k \approx \arg \min_{\|s\| \leq \Delta_k} m_k(s) := \nabla F(x_k^{(k)})^T s + \frac{1}{2} s^T H(x_k^{(k)}) s$

4: $\rho_k := (F(x_k^{(k)}) + s_k) - F(x_k^{(k)}))/m_k(s_k)$

5: if $\rho_k \geq \eta$ then

6: $x^{(k+1)} = x^{(k)} + s_k$ and $\Delta_{k+1} = \gamma \Delta_k$

7: else

8: $x^{(k+1)} = x^{(k+1)}$ and $\Delta_{k+1} = \gamma^{-1} \Delta_k$

9: end if

10: end for
**Theorem (Complexity of Stochastic TR)**

If $\epsilon \in \mathcal{O}(\epsilon_H)$, then Stochastic TR terminates after

$$T \in \mathcal{O} \left( \max \{ \epsilon_g^{-2} \epsilon_H^{-1}, \epsilon_H^{-3} \} \right),$$

iterations, upon which, with high probability, we have that

$$\| \nabla F(x) \| \leq \epsilon_g,$$  

and  

$$\lambda_{\min}(\nabla^2 F(x)) \geq - (\epsilon + \epsilon_H).$$

- This is tight!
Algorithm 7 Stochastic Adaptive Regularization with Cubic Algorithm

1: \textbf{Input:} $x_0$, $\Delta_0 > 0$, $\eta \in (0, 1)$, $\gamma > 1$, $0 < \epsilon, \epsilon_g, \epsilon_H < 1$
2: \textbf{for} $k = 0, 1, 2, \cdots$ \textbf{until} termination \textbf{do}
3:  
   \begin{align*}
   s_k &\approx \arg\min_{s \in \mathbb{R}^d} m_k(s) := \nabla F(x_k^{(k)})^T s + \frac{1}{2} s^T H(x_k^{(k)}) s + \frac{\delta_k}{3} \|s\|^3 \\
   \end{align*}
4:  
   $\rho_k := (F(x_k^{(k)} + s_k) - F(x_k^{(k)})) / m_k(s_k)$.
5:  
   \textbf{if} $\rho_k \geq \eta$ \textbf{then}
6:  
   \begin{align*}
   x^{(k+1)} & = x^{(k)} + s_k \quad \text{and} \quad \sigma_{k+1} = \gamma^{-1} \Delta_k \\
   \end{align*}
7:  
   \textbf{else}
8:  
   \begin{align*}
   x^{(k+1)} & = x^{(k+1)} \quad \text{and} \quad \sigma_{k+1} = \gamma \Delta_k \\
   \end{align*}
9:  
   \textbf{end if}
10: \textbf{end for}
Theorem (Complexity of Stochastic ARC)

If $\epsilon \in \mathcal{O}(\epsilon_g, \epsilon_H)$, then Stochastic TR terminates after

$$T \in \mathcal{O}\left(\max\{\epsilon_g^{-3/2}, \epsilon_H^{-3}\}\right),$$

iterations, upon which, with high probability, we have that

$$\|\nabla F(x)\| \leq \epsilon_g, \quad \text{and} \quad \lambda_{\min}(\nabla^2 F(x)) \geq - (\epsilon + \epsilon_H).$$

This is tight!
For $\epsilon_H^2 = \epsilon_g = \epsilon = \epsilon_0$

- **Stochastic TR**: $T \in \mathcal{O}(\epsilon_0^{-3})$
- **Stochastic ARC**: $T \in \mathcal{O}(\epsilon_0^{-3/2})$
**Non-Linear Least Squares**

\[
\min_{x \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \left( b_i - \Phi(a_i^T x_i) \right)^2
\]
NON-LINEAR LEAST SQUARES: SYNTHETIC, $n = 1000,000, d = 1000, s = 1\%$

(a) Train Loss vs. Time
(b) Train Loss vs. Time
Conclusions: Second order machine learning

- Second order methods
  - A simple way to go beyond first order methods
  - Obviously, don’t be naïve about the details

- FLAG n’ FLARE
  - Combine acceleration and adaptivity to get best of both worlds

- Can aggressively sub-sample gradient and/or Hessian
  - Improve running time at each step
  - Maintain strong second-order convergence

- Apply to non-convex problems
  - Trust region methods and cubic regularization methods
  - Converge to second order stationary point