Collaborators

- Peng Xu (Stanford)
- Jiyan Yang (Stanford)
- Christopher Ré (Stanford)
- Michael Mahoney (ICSI, UCB)
WE’VE DECIDED TO TAKE BIG DATA TO THE NEXT LEVEL...

HUMONGOUS DATA

© D.Fletcher for CloudTweaks.com
**Problem Statement**

Let \( x \in \mathcal{X} \subseteq \mathbb{R}^p \), the problem is to minimize \( F(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x) \)

- \( f_i \Rightarrow \) smooth
- \( n \gg 1 \) and/or \( p \gg 1 \)
Examples

- Machine Learning and Data fitting
  - $f_i \Rightarrow$ loss corresponding to $i^{th}$ observation
    - GLM, SVM, Neural Networks, and many more …
Examples

- Linear/Nonlinear Inverse Problems
  - $f_i \Rightarrow$ misfit corresponding to $i^{th}$ measurement
  - e.g., PDE Inverse Problems: $f_i(x) = \frac{1}{2} \| P_i A^{-1}(x) q_i - d_i \|^2_2$
Modern "Humongous-Data"

- Classical Algorithms $\Rightarrow$ High Per-Iteration Cost
Modern “Humongous-Data”

1. Need to design variants, that are:

   1. Low Per-Iteration Cost
   2. Fast Convergence Rate
First Order Methods

- Use only gradient information
  - E.g.: Gradient Descent

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

- Smooth Convex \( F \Rightarrow \) Sublinear, \( O(1/k) \)
- Smooth Strongly Convex \( F \Rightarrow \) Linear, \( O(\rho^k), \rho < 1 \)
- However, iteration cost scales linearly in \( n \)
First Order Methods

- **Stochastic** variants e.g., (mini-batch) SGD
  - $S \subset \{1, 2, \ldots, n\}$ is chosen at random with $|S| \ll n$
  
  $$x^{(k+1)} = x^{(k)} - \frac{\alpha_k}{|S|} \sum_{j \in S} \nabla f_j(x^{(k)})$$

- Cheap per-iteration costs!
- However slower to converge:
  - Smooth Convex $F \Rightarrow O(1/\sqrt{k})$
  - Smooth Strongly Convex $F \Rightarrow O(1/k)$
• Modifications...
  • Achieve the convergence rate of the full GD
  • Preserve the per-iteration cost of SGD
• E.g.: SAG, SDCA, SVRG, S2GD, Acc-Prox-SDCA, Prox-SVRG, MISO, SAGA, Acc-Prox-SVRG, mS2GD, AMSVRG,...
1st Order Method
ML Paradox

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

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

- Avoids Over-fitting? i.e.,
ML Paradox

\[(a, b) \sim \mathcal{D} \text{ and } \ell \text{ is loss} \]

\[h^* = \arg \min_{h \in \mathcal{H}} \mathbb{E}_\mathcal{D} \ell(h(a), b) \quad \text{“True Risk Minimization”} \]
ML Paradox

\((a, b) \sim \mathcal{D} \text{ and } \ell \text{ is loss} \)

\[ h^* = \arg \min_{h \in \mathcal{H}} \mathbb{E}_\mathcal{D} \ell(h(a), b) \quad \text{“True Risk Minimization”} \]

\[ \hat{h} = \arg \min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(h(a_i), b_i) \quad \text{“Empirical Risk Minimization”} \]
ML Paradox

\((a, b) \sim \mathcal{D}\) and \(\ell\) is loss

\[ h^* = \arg\min_{h \in \mathcal{H}} \mathbb{E}_D \ell(h(a), b) \quad \text{“True Risk Minimization”} \]

\[ \hat{h} = \arg\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(h(a_i), b_i) \quad \text{“Empirical Risk Minimization”} \]

\[ \mathbb{E}_D \ell(\hat{h}(a), b) \leq \mathbb{E}_D \ell(h^*(a), b) + \mathcal{O}\left(\sqrt{\frac{\text{VC}(\mathcal{H})}{n}}\right) \quad \text{w.h.p} \]
Q: If $n \gg 1$, why do we use (stochastic) 1st order methods?

- Cheaper Iterations? ✓
- Avoids Over-fitting? ✗ ⇒ We do want minimizer!!!
Second Order Methods

- Use both gradient and Hessian information

- E.g.: Newton’s method

\[ x^{(k+1)} = x^{(k)} - \left[ \nabla^2 F(x^{(k)}) \right]^{-1} \nabla F(x^{(k)}) \]

- Smooth Convex \( F \) ⇒ Locally Superlinear
- Smooth Strongly Convex \( F \) ⇒ Locally Quadratic
- However, per-iteration cost is high!
Second Order Methods

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

- 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]
Let’s “make Newton method great again”!!!
Iterative Scheme

\[ x^{(k+1)} = \arg \min_{x \in D \cap X} \left\{ F(x^{(k)}) + (x - x^{(k)})^T g(x^{(k)}) + \frac{1}{2\alpha_k} (x - x^{(k)})^T H(x^{(k)}) (x - x^{(k)}) \right\} \]
Iterative Scheme

\[ x^{(k+1)} = \arg \min_{x \in \mathcal{D} \cap X} \left\{ F(x^{(k)}) + (x - x^{(k)})^T g(x^{(k)}) + \frac{1}{2\alpha_k} (x - x^{(k)})^T H(x^{(k)})(x - x^{(k)}) \right\} , \]

- **Newton:** \[ g(x^{(k)}) = \nabla F(x^{(k)}) \quad \& \quad H(x^{(k)}) = \nabla^2 F(x^{(k)}) \]
- **Projected Gradient Descent:** \[ g(x^{(k)}) = \nabla F(x^{(k)}) \quad \& \quad H(x^{(k)}) = \mathbb{I} \]
- **Frank-Wolfe:** \[ g(x^{(k)}) = \nabla F(x^{(k)}) \quad \& \quad H(x^{(k)}) = 0 \]
- **(mini-batch) SGD:** \[ g(x^{(k)}) = \frac{1}{|S_g|} \sum_{j \in S_g} \nabla f_j(x^{(k)}) \quad \& \quad H(x^{(k)}) = \mathbb{I} \]
- **Sub-Sampled Newton (SSN):**
  - **SSN w. Hessian Sub-Sampling:**
    \[ g(x^{(k)}) = \nabla F(x^{(k)}) \quad \& \quad H(x^{(k)}) = \frac{1}{|S_H|} \sum_{j \in S_H} \nabla^2 f_i(x^{(k)}) \]
  - **SNN w. Gradient and Hessian Sub-Sampling:**
    \[ g(x^{(k)}) = \frac{1}{|S_g|} \sum_{j \in S_g} \nabla f_j(x^{(k)}) \quad \& \quad H(x^{(k)}) = \frac{1}{|S_H|} \sum_{j \in S_H} \nabla^2 f_i(x^{(k)}) \]
Sub-Sampled Newton methods:

- Globally Convergent Algorithms [RM-I, 2016]
  - Approach the optimum, \( x^* \), from any \( x^{(0)} \)
  - Uniform Sampling

- Local Convergence Rate [RM-II, 2016], [PYRRM, 2016]
  - Achieve fast rate, at least locally
  - Uniform and non-uniform sampling
Hessian Sub-Sampling

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

\[ H(x) = \frac{1}{|S|} \sum_{j \in S} \nabla^2 f_j(x) \]
\[ \nabla^2 f_i(x) \preceq K I, \quad \forall x \in \mathbb{R}^p, \quad i = 1, 2, \ldots, n \]
\[ \| \nabla^2 f_i(x) - \nabla^2 f_i(y) \| \leq L \| x - y \|, \quad \forall x, y \in \mathbb{R}^p, \quad i = 1, 2, \ldots, n \]
\[ \nabla^2 F(x) \succeq \gamma I, \quad \forall x \in \mathbb{R}^p. \]

- “Sub-Sampling” Condition Number: \( \kappa = \frac{K}{\gamma} \)
“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
“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 learning.”

Prof. Jorge Nocedal
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**: 
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:** 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 $x \in \mathbb{R}^p$, if

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

then

$$\Pr \left( \left| \lambda_i \left( \nabla^2 F(x) \right) - \lambda_i \left( H(x) \right) \right| \leq \epsilon \lambda_i \left( \nabla^2 F(x) \right); \ i = 1, 2, \cdots, p \right) \geq 1 - \delta.$$
**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)**

Let \(0 < \delta < 1\) and \(0 < \epsilon < 1\) be given. Using \(\alpha_k = 1\), with probability \(1 - \delta\), 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!
Algorithm 1 Locally Convergent SSN-H with exact solve

1: **Input:** $\mathbf{x}^{(0)}$, $0 < \delta < 1$, $0 < \epsilon < 1$
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 $H(\mathbf{x}^{(k)})$
5: - Update $\mathbf{x}^{(k+1)}$ with $H(\mathbf{x}^{(k)})$ and $\alpha_k = 1$
6: end for
Consider any $0 < \rho_0 < \rho < 1$ and $\epsilon \leq \rho_0/(1 + \rho_0)$. If
\[ \|\mathbf{x}(0) - \mathbf{x}^*\| \leq \frac{\rho - \rho_0}{\xi}, \]
using Algorithm 1, we get locally Q-linear convergence
\[ \|\mathbf{x}(k) - \mathbf{x}^*\| \leq \rho \|\mathbf{x}(k-1) - \mathbf{x}^*\|, \quad k = 1, \ldots, k_0 \]
with probability $(1 - \delta)^{k_0}$. 
Using Algorithm 1, with

\[ \epsilon^{(k)} = \rho^k \epsilon, \quad k = 0, 1, \ldots, k_0, \]

if \( x^{(0)} \) is close-enough, we get locally Q-superlinear convergence

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

with probability \((1 - \delta)^{k_0}\).
Theorem (Q-Superlinear Convergence: Slow Growth)

Using Algorithm 1 with

$$\epsilon^{(k)} = \frac{1}{1 + 2 \ln(4 + k)}, \quad k = 0, 1, \ldots, k_0,$$

if $x^{(0)}$ is close-enough, we get locally Q-superlinear convergence

$$\|x^{(k)} - x^*\| \leq \frac{1}{\ln(3 + k)} \|x^{(k-1)} - x^*\|, \quad k = 1, \ldots, k_0,$$

with probability $(1 - \delta)^{k_0}$. 

Fred Roosta (ICSI)
**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
Possible to make it globally convergent with e.g., Armijo line search: Find largest $\alpha_k \leq 1$

$$F(x^{(k)} + \alpha_k p_k) \leq F(x^{(k)}) + \alpha_k \beta p_k^T \nabla F(x^{(k)})$$

$$x^{(k+1)} = x^{(k)} + \alpha_k p_k$$
Consider any $0 < \rho_0 < \rho < 1$. Using Algorithm 1 with line search, any $x^{(0)} \in \mathbb{R}^p$, $0 < \beta < 1/2$ and

$$\epsilon \leq \min \left\{ \frac{(1 - 2\beta)}{2(1 - \beta)}, \frac{\rho_0}{4(1 + \rho_0)\sqrt{\kappa}} \right\},$$

after $O(\kappa^2)$ iterations, with probability $(1 - \delta)^k$ we get “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 unite step length [accidentally] works.”

Prof. Jorge Nocedal
**Simulations: $\ell_2$ Regularized LR**

$$F(x) = \frac{1}{n} \sum_{i=1}^{n} \left( \ln \left( 1 + \exp(a_i^T x) \right) - b_i a_i^T x \right) + \frac{\lambda}{2} \|x\|^2$$

<table>
<thead>
<tr>
<th>Data</th>
<th>$n$</th>
<th>$p$</th>
<th>NNZ</th>
<th>$\kappa(F)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_1$</td>
<td>$10^6$</td>
<td>$10^4$</td>
<td>0.02%</td>
<td>$\approx 10^4$</td>
</tr>
<tr>
<td>$D_2$</td>
<td>$5 \times 10^4$</td>
<td>$5 \times 10^3$</td>
<td>DENSE</td>
<td>$\approx 10^6$</td>
</tr>
<tr>
<td>$D_3$</td>
<td>$10^7$</td>
<td>$2 \times 10^4$</td>
<td>0.006%</td>
<td>$\approx 10^{10}$</td>
</tr>
</tbody>
</table>
$D_1, n = 10^6, p = 10^4, \text{SPARSITY} : 0.02\%, \kappa \approx 10^4$
$D_2, n = 5 \times 10^4, p = 5 \times 10^3, \text{sparsity : Dense, } \kappa \approx 10^6$
$D_3, n = 10^7, p = 2 \times 10^4, \text{SPARSITY : 0.006\%}, \kappa \approx 10^{10}$
Extreme non-uniformity $\Rightarrow O(n)$ uniform samples!!!
Better sampling strategy:

- $|S|$ independent of $n$
- Immune non-uniformity

Can we construct such sampling scheme?
**Non-Uniform**

- $\nabla^2 f_i = A_i^T A_i, \quad A_i \in \mathbb{R}^{k_i \times p}$
- E.g., $f_i = \ell(a_i^T x) \Rightarrow A_i = \sqrt{\ell''(a_i^T x)} a_i^T$
Non-Uniform Sampling Schemes [XYRRM, 2016]:

- Block Norm Squares
- Block Partial Leverage Scores
Table: $A = [A_1^T A_2^T \ldots A_n^T]^T$

<table>
<thead>
<tr>
<th>Name</th>
<th>Complexity per iteration</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Newton-CG</td>
<td>$\mathcal{O}(\text{NNZ}(A) \sqrt{\kappa})$</td>
<td>Folklore</td>
</tr>
<tr>
<td>SSN-LS</td>
<td>$\tilde{\mathcal{O}}(\text{NNZ}(A) \log n + d^2 \kappa^{3/2})$</td>
<td>[XYRRM, 2016]</td>
</tr>
<tr>
<td>SSN-RNS</td>
<td>$\tilde{\mathcal{O}}(\text{NNZ}(A) + \text{sr}(A)d\kappa^{5/2})$</td>
<td>[XYRRM, 2016]</td>
</tr>
<tr>
<td>SRHT</td>
<td>$\tilde{\mathcal{O}}(nd(\log n)^4 + d^2(\log n)^4 \kappa^{3/2})$</td>
<td>[PILANCHI ET AL., 2016]</td>
</tr>
<tr>
<td>SSN-Uniform</td>
<td>$\tilde{\mathcal{O}}(\text{NNZ}(A) + d\hat{\kappa}\kappa^{3/2})$</td>
<td>[RM, 2016]</td>
</tr>
<tr>
<td>LiSSA</td>
<td>$\tilde{\mathcal{O}}(\text{NNZ}(A) + d\bar{\kappa}\kappa^{3/2})$</td>
<td>[AGRAWAL ET AL., 2016]</td>
</tr>
</tbody>
</table>

\[
\kappa = \max_x \frac{\lambda_{\max} \nabla^2 F(x)}{\lambda_{\min} \nabla^2 F(x)} \\
\hat{\kappa} = n \cdot \max_x \frac{\max_i \lambda_{\max} \nabla^2 f_i(x)}{\lambda_{\min} \nabla^2 F(x)} \\
\bar{\kappa} = \max_x \frac{\max_i \lambda_{\max} \nabla^2 f_i(x)}{\min_i \lambda_{\min} \nabla^2 f_i(x)}
\]

\[
\Rightarrow \kappa \leq \hat{\kappa} \leq \bar{\kappa}
\]
THANK YOU!