SSN: Sub-Sampled Newton Methods

Fred Roosta

ICSI
&
Department of Statistics
UC Berkeley
June, 2016

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

Problem

$$\min_{\mathbf{x}\in\mathcal{X}\subseteq\mathbb{R}^p} F(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x})$$

- $f_i \Rightarrow$ 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: $f_i(\mathbf{x}) = \frac{1}{2}\|P_i A^{-1}(\mathbf{x})\mathbf{q}_i - \mathbf{d}_i\|_2^2$

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

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \nabla F(\mathbf{x}^{(k)})$$

  - Smooth Convex $F \Rightarrow$ Sublinear, $\mathcal{O}(1/k)$
  - Smooth Strongly Convex $F \Rightarrow$ Linear, $\mathcal{O}(\rho^k)$, $\rho < 1$
  - However, iteration cost scales linearly in n

First Order Methods

- Stochastic variants e.g., (mini-batch) SGD
  - $\mathcal{S} \subset \{1, 2, \cdots, n\}$ is chosen at random with $|\mathcal{S}| \ll n$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \frac{\alpha_k}{|\mathcal{S}|} \sum_{j \in \mathcal{S}} \nabla f_j(\mathbf{x}^{(k)})$$

  - Cheap per-iteration costs!
  - However slower to converge:
    - Smooth Convex $F \Rightarrow \mathcal{O}(1/\sqrt{k})$
    - Smooth Strongly Convex $F \Rightarrow \mathcal{O}(1/k)$

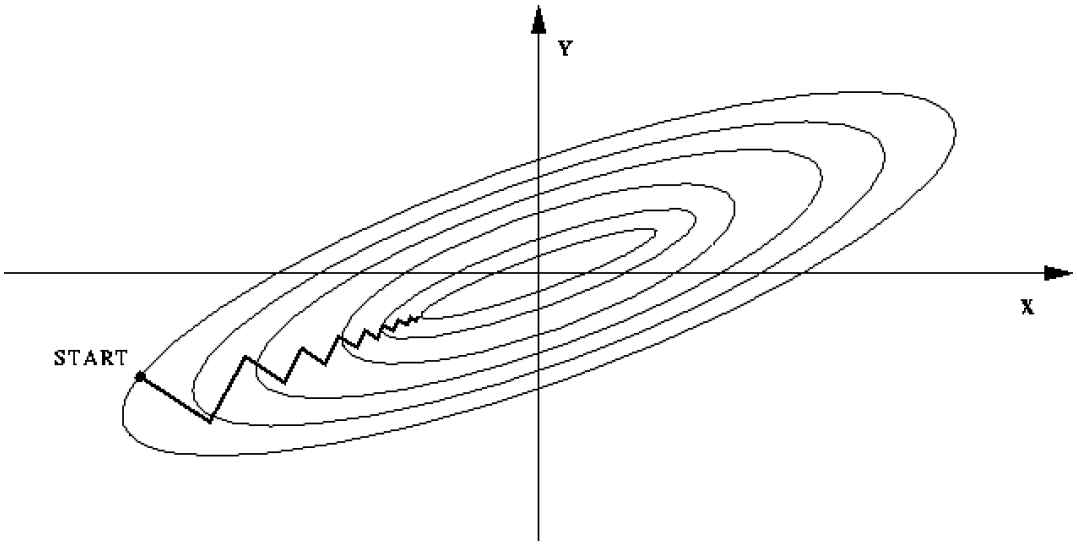
- Modifications...
  - Achieve the convergence rate of the full GD
  - Preserve the per-iteration cost of SGD
- E.g.: SAG, SDCA, SVRG, S2GD, Acc-Prox-SDCA, Prox-SVRG, MISO, SAGA, Acc-Prox-SVRG, mS2GD, AMSVRG,...


ML
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

$(\mathbf{a}, b) \sim \mathcal{D}$ and ℓ is loss

$$h^* = \arg\min_{h \in \mathcal{H}} \mathbb{E}_{\mathcal{D}} \ell(h(\mathbf{a}), b) \quad \text{"True Risk Minimization"}$$

ML Paradox

$(\mathbf{a}, b) \sim \mathcal{D}$ and ℓ is loss

$$h^* = \arg\min_{h \in \mathcal{H}} \mathbb{E}_{\mathcal{D}} \ell(h(\mathbf{a}), b) \quad \text{"True Risk Minimization"}$$

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(h(\mathbf{a}_i), b_i) \quad \text{"Empirical Risk Minimization"}$$

ML Paradox

$(\mathbf{a}, b) \sim \mathcal{D}$ and ℓ is loss

$$h^* = \arg\min_{h \in \mathcal{H}} \mathbb{E}_{\mathcal{D}} \ell(h(\mathbf{a}), b) \quad \text{"True Risk Minimization"}$$

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(h(\mathbf{a}_i), b_i) \quad \text{"Empirical Risk Minimization"}$$

$$\mathbb{E}_{\mathcal{D}} \ell(\hat{h}(\mathbf{a}), b) \leq \mathbb{E}_{\mathcal{D}} \ell(h^*(\mathbf{a}), b) + \mathcal{O}\left(\sqrt{\frac{VC(\mathcal{H})}{n}}\right) \quad \text{w.h.p}$$

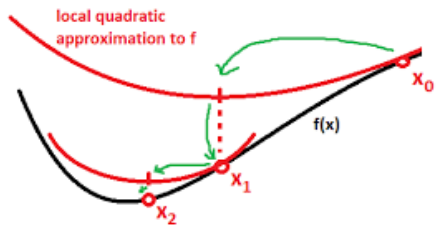
ML Paradox

Q: If $n \gg 1$, why do we use (stochastic) 1st order methods?

- Cheaper Iterations? ✓
- Avoids Over-fitting? ✗ $\Rightarrow$ We do want minimizer!!!

Second Order Methods

- Use both gradient and Hessian information
  - E.g. : Newton's method



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

- Smooth Convex $F \Rightarrow$ Locally Superlinear
- Smooth Strongly Convex $F \Rightarrow$ 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]

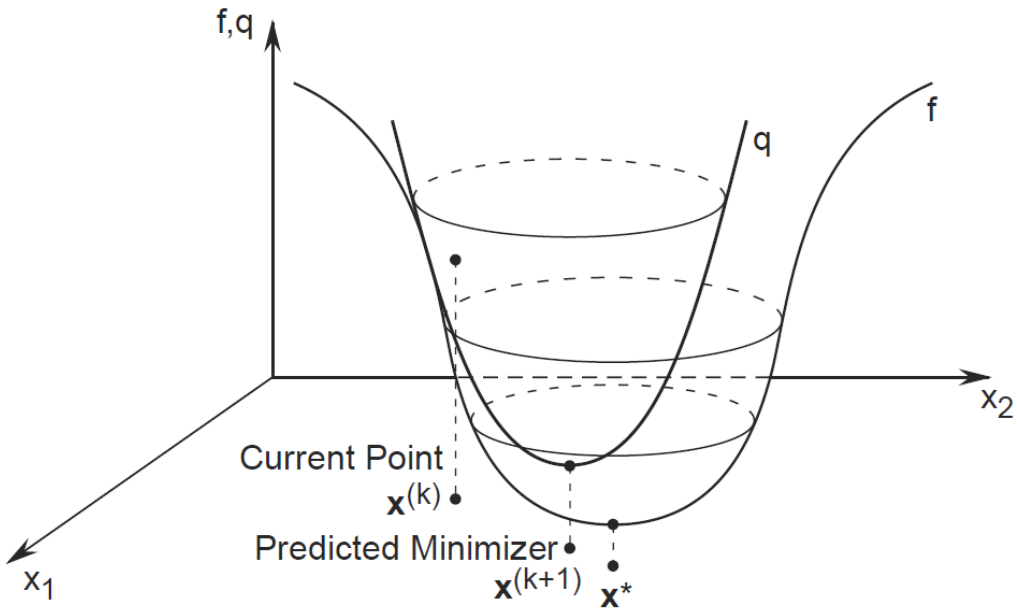
Second Order Methods

- Randomly approximating second order information cheaply
  - Sketching the Hessian [Pilanci et al., 2015]
  - Sub-Sampling the Hessian [Byrd et al., 2011, Erdogdu et al., 2015, Martens, 2010, RM-I, RM-II, XYRRM, 2016]
  - Sub-Sampling the Hessian and the gradient [RM-I & RM-II, 2016]

Let's "make Newton method great again"!!!

Iterative Scheme

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



Iterative Scheme

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

- Newton: $\mathbf{g}(\mathbf{x}^{(k)}) = \nabla F(\mathbf{x}^{(k)}) \quad \& \quad H(\mathbf{x}^{(k)}) = \nabla^2 F(\mathbf{x}^{(k)})$
- Projected Gradient Descent: $\mathbf{g}(\mathbf{x}^{(k)}) = \nabla F(\mathbf{x}^{(k)}) \quad \& \quad H(\mathbf{x}^{(k)}) = \mathbb{I}$
- Frank-Wolfe: $\mathbf{g}(\mathbf{x}^{(k)}) = \nabla F(\mathbf{x}^{(k)}) \quad \& \quad H(\mathbf{x}^{(k)}) = 0$
- (mini-batch) SGD: $\mathbf{g}(\mathbf{x}^{(k)}) = \frac{1}{|\mathcal{S}_\mathbf{g}|} \sum_{j \in \mathcal{S}_\mathbf{g}} \nabla f_j(\mathbf{x}^{(k)}) \quad \& \quad H(\mathbf{x}^{(k)}) = \mathbb{I}$
- Sub-Sampled Newton (SSN):
  - SSN w. Hessian Sub-Sampling:

$$\mathbf{g}(\mathbf{x}^{(k)}) = \nabla F(\mathbf{x}^{(k)}) \quad \& \quad H(\mathbf{x}^{(k)}) = \frac{1}{|\mathcal{S}_H|} \sum_{j \in \mathcal{S}_H} \nabla^2 f_i(\mathbf{x}^{(k)})$$

  - SNN w. Gradient and Hessian Sub-Sampling:

$$\mathbf{g}(\mathbf{x}^{(k)}) = \frac{1}{|\mathcal{S}_\mathbf{g}|} \sum_{j \in \mathcal{S}_\mathbf{g}} \nabla f_j(\mathbf{x}^{(k)}) \quad \& \quad H(\mathbf{x}^{(k)}) = \frac{1}{|\mathcal{S}_H|} \sum_{j \in \mathcal{S}_H} \nabla^2 f_i(\mathbf{x}^{(k)})$$

SSN

- Sub-Sampled Newton methods:
  - Globally Convergent Algorithms [RM-I, 2016]
    - Approach the optimum, $\mathbf{x}^*$, from any $\mathbf{x}^{(0)}$
    - Uniform Sampling
  - Local Convergence Rate [RM-II, 2016], [PYRRM, 2016]
    - Achieve fast rate, at least locally
    - Uniform and non-uniform sampling

Hessian Sub-Sampling

$$\mathbf{g}(\mathbf{x}) = \nabla F(\mathbf{x})$$

$$H(\mathbf{x}) = \frac{1}{|\mathcal{S}|} \sum_{j \in S} \nabla^2 f_j(\mathbf{x})$$

Assumptions

$$\nabla^2 f_i(\mathbf{x}) \preceq K\mathbb{I}, \quad \forall \mathbf{x} \in \mathbb{R}^p, \quad i = 1, 2, \ldots, n$$

$$\|\nabla^2 f_i(\mathbf{x}) - \nabla^2 f_i(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p, \quad i = 1, 2, \ldots, n$$

$$\nabla^2 F(\mathbf{x}) \succeq \gamma\mathbb{I}, \quad \forall \mathbf{x} \in \mathbb{R}^p.$$

- "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 leaning."

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:** $|\mathcal{S}|$ 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:** $|\mathcal{S}|$ 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:** $H(\mathbf{x})$ must preserve the spectrum of $\nabla^2 F(\mathbf{x})$ as much as possible

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

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

What do we need?

- Requirements:

(R.1) **Scale up:** $|\mathcal{S}|$ 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:** $H(\mathbf{x})$ must preserve the spectrum of $\nabla^2 F(\mathbf{x})$ 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:** $|\mathcal{S}|$ 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:** $H(\mathbf{x})$ must preserve the spectrum of $\nabla^2 F(\mathbf{x})$ 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

Sub-sampling Hessian

- Requirements:

(R.1) **Scale up:** $|\mathcal{S}|$ 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:** $H(\mathbf{x})$ must preserve the spectrum of $\nabla^2 F(\mathbf{x})$ 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

Sub-sampling Hessian

Lemma (Uniform Hessian Sub-Sampling)

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

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

then

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

Sub-sampling Hessian

- Requirements:

(R.1) **Scale up:** $|\mathcal{S}|$ 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:** $H(\mathbf{x})$ must preserve the spectrum of $\nabla^2 F(\mathbf{x})$ 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

Error Recursion: Hessian Sub-Sampling

Theorem (Error Recursion)

Let $0 < \delta < 1$ *and* $0 < \epsilon < 1$ *be given. Using* $\alpha_k = 1$ *, with probability* $1 - \delta$*, we have*

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

where

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

- ρ_0 is problem-independent! $\Rightarrow$ Can be made arbitrarily small!

SSN-H Algorithm

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, $|\mathcal{S}|$, with ϵ and δ
3: **for** $k = 0, 1, 2, \cdots$ until termination **do**
4: - Select a sample set, $\mathcal{S}$, of size $|\mathcal{S}|$ and $H(\mathbf{x}^{(k)})$
5: - Update $\mathbf{x}^{(k+1)}$ with $H(\mathbf{x}^{(k)})$ and $\alpha_k = 1$
6: **end for**

SSN-H: Q-Linear Convergence

Theorem (Q-Linear Convergence)

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

SSN-H: Q-Superinear Convergence

Theorem (Q-Superlinear Convergence: Geometric Growth)

Using Algorithm 1, with

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

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

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

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

Hessian Sub-Sampling: Q-Superinear Convergence

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

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

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

Sub-sampling Hessian

- Requirements:

(R.1) **Scale up:** $|\mathcal{S}|$ 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:** $H(\mathbf{x})$ must preserve the spectrum of $\nabla^2 F(\mathbf{x})$ 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

Global

Possible to make it globally convergent with e.g., Armijo line search: Find largest $\alpha_k \leq 1$

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

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{p}_k$$

Local + Global

Theorem

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

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

after $\mathcal{O}(\kappa^2)$ *iterations, with probability* $(1-\delta)^k$ *we get "problem-independent" Q-linear convergence, i.e.,*

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

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."

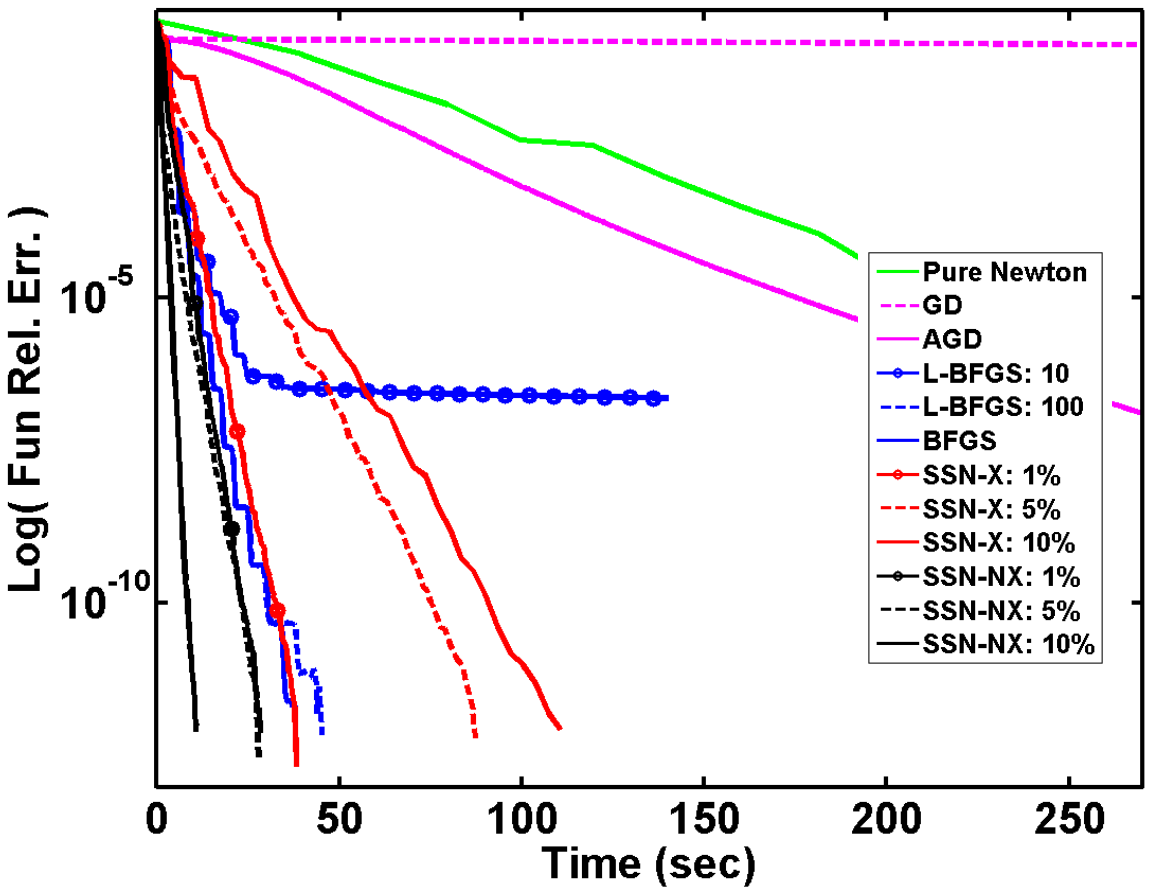
Prof. Jorge Nocedal

Simulations: ℓ_2 Regularized LR

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

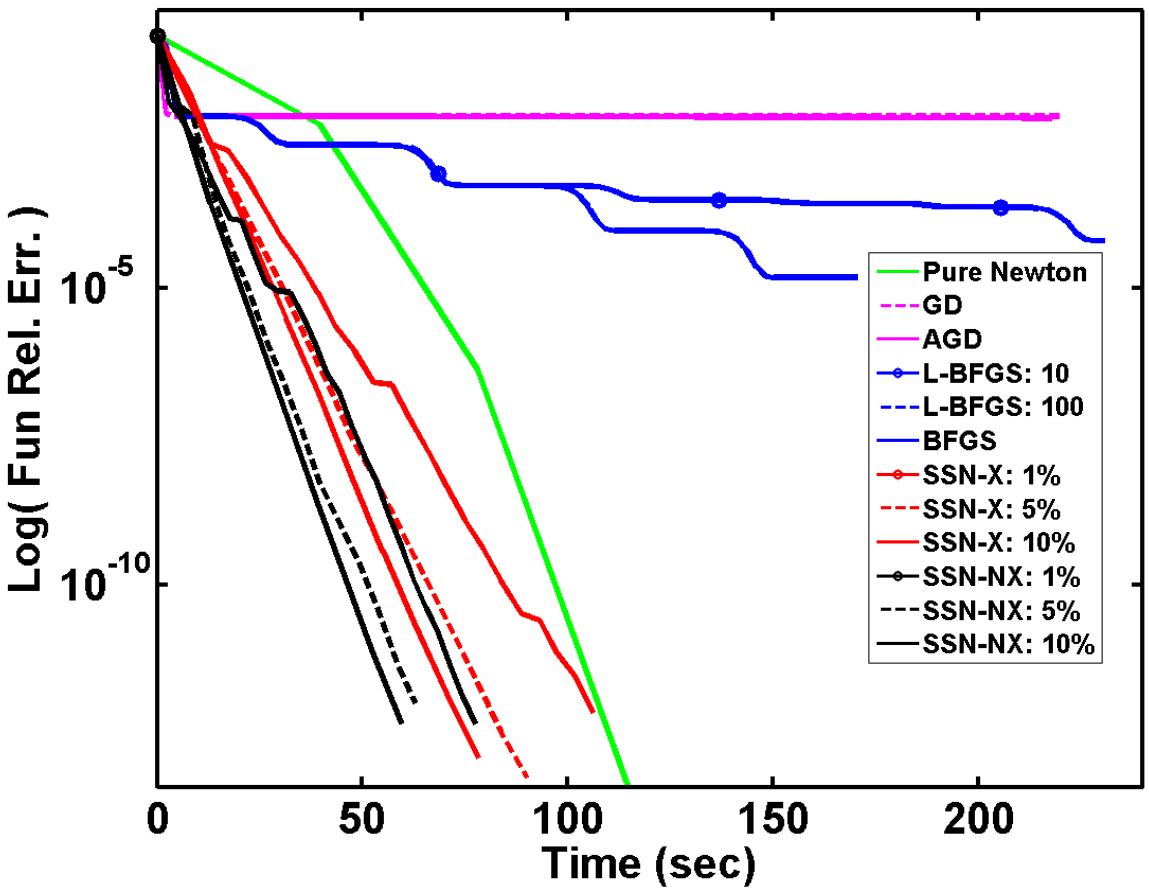
Data	n	p	nnz	$\kappa(F)$
D_1	10^6	10^4	0.02%	$\approx 10^4$
D_2	5×10^4	5×10^3	Dense	$\approx 10^6$
D_3	10^7	2×10^4	0.006%	$\approx 10^{10}$

$D_1, n = 10^6, p = 10^4$, SPARSITY : 0.02%, $\kappa \approx 10^4$



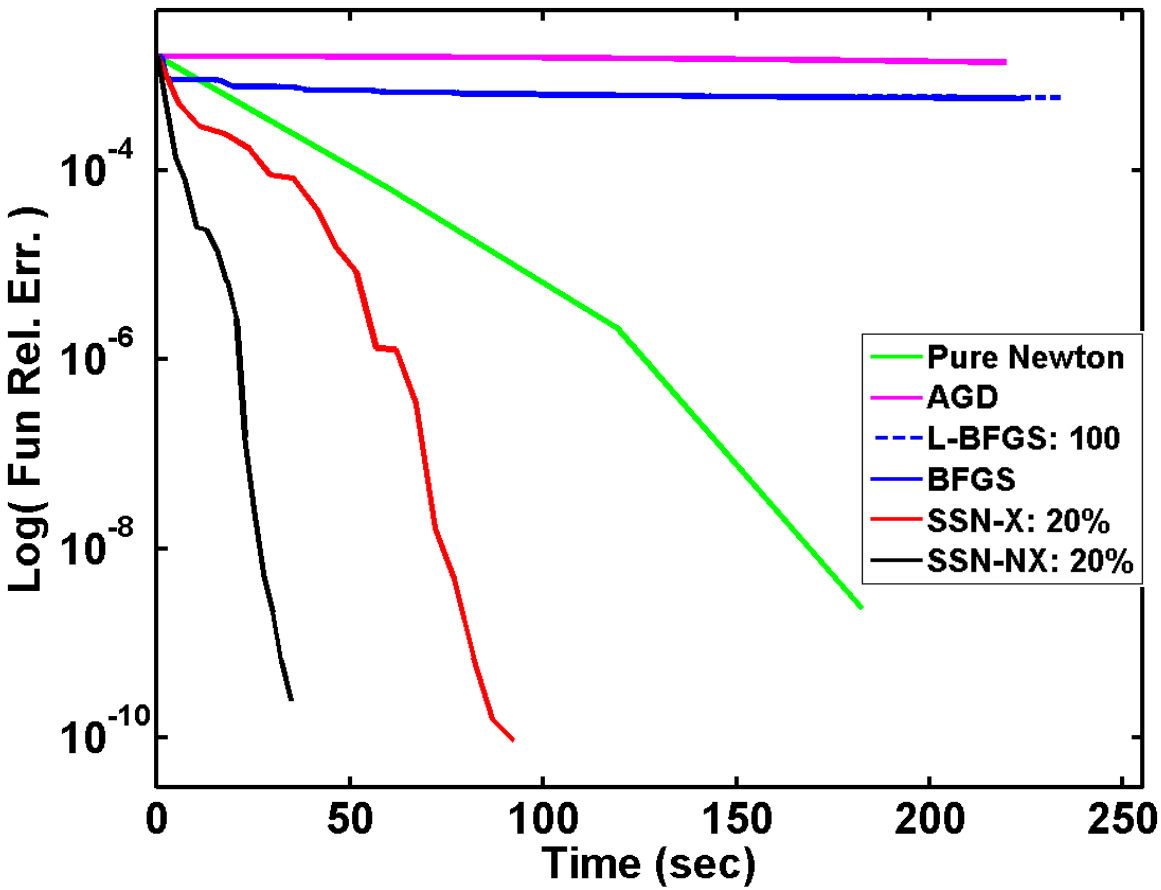
(a) Function Relative Error

$D_2, n = 5 \times 10^4, p = 5 \times 10^3$, SPARSITY : DENSE, $\kappa \approx 10^6$



(c) Function Relative Error

$D_3, n = 10^7, p = 2 \times 10^4, \text{SPARSITY} : 0.006\%, \kappa \approx 10^{10}$



(e) Function Relative Error

Extreme non-uniformity $\Rightarrow$ $\mathcal{O}(n)$ uniform samples!!!

Non-Uniform

Better sampling strategy:

- $|\mathcal{S}|$ 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., $f_i = \ell(\mathbf{a}_i^T \mathbf{x}) \Rightarrow A_i = \sqrt{\ell''(\mathbf{a}_i^T \mathbf{x})}\mathbf{a}_i^T$

- Non-Uniform Sampling Schemes [XYRRM, 2016]:
  - Block Norm Squares
  - Block Partial Leverage Scores

TABLE: $A = [A_1^T A_2^T \dots A_n^T]^T$

NAME	COMPLEXITY PER ITERATION	REFERENCE
NEWTON-CG	$\mathcal{O}(\text{NNZ}(A)\sqrt{\kappa})$	FOLKLORE
SSN-LS	$\tilde{\mathcal{O}}(\text{NNZ}(A)\log n + d^2\kappa^{3/2})$	[XYRRM, 2016]
SSN-RNS	$\tilde{\mathcal{O}}(\text{NNZ}(A) + \mathbf{sr}(A)d\kappa^{5/2})$	[XYRRM, 2016]
SRHT	$\tilde{\mathcal{O}}(nd(\log n)^4 + d^2(\log n)^4\kappa^{3/2})$	[PILANCI ET AL., 2016]
SSN-UNIFORM	$\tilde{\mathcal{O}}(\text{NNZ}(A) + d\hat{\kappa}\kappa^{3/2})$	[RM, 2016]
LISSA	$\tilde{\mathcal{O}}(\text{NNZ}(A) + d\hat{\kappa}\bar{\kappa}^2)$	[AGRAWAL ET AL., 2016]

$$\left.\begin{aligned} \kappa &= \max_{\mathbf{x}} \frac{\lambda_{\max}\nabla^2 F(\mathbf{x})}{\lambda_{\min}\nabla^2 F(\mathbf{x})} \\ \hat{\kappa} &= n \cdot \max_{\mathbf{x}} \frac{\max_i \lambda_{\max}\nabla^2 f_i(\mathbf{x})}{\lambda_{\min}\nabla^2 F(\mathbf{x})} \\ \bar{\kappa} &= \max_{\mathbf{x}} \frac{\max_i \lambda_{\max}\nabla^2 f_i(\mathbf{x})}{\min_i \lambda_{\min}\nabla^2 f_i(\mathbf{x})} \end{aligned}\right\} \Rightarrow \kappa \leq \hat{\kappa} \leq \bar{\kappa}$$

THANK YOU!