Sub-sampled Newton Methods with Non-uniform Sampling

Jiyan Yang

ICME, Stanford University

IAS/PCMI Research Program, July 14, 2016
Joint work with Peng Xu, Fred Roosta, Chris Ré and Michael Mahoney

Problem formulation

- Consider the optimization problem

$$\min_{\mathbf{w}\in\mathcal{C}} F(\mathbf{w}) = \sum_{i=1}^{n} f_i(\mathbf{w}) + R(\mathbf{w}), \tag{1}$$

  where $f_i(\mathbf{w})$ and $R(\mathbf{w})$ are convex and twice-differentiable (assume $\mathcal{C} = \mathbb{R}^d$ in this talk)

- Example:

$$f_i(\mathbf{w}) = \ell(\mathbf{x}_i^T\mathbf{w}), \quad R(\mathbf{w}) = \frac{\lambda}{2}\|\mathbf{w}\|_2^2, \tag{2}$$

  where $\ell(\cdot)$ is a loss function and $\mathbf{x}_i$'s are data points

Second-order methods

- There is a plethora of first-order optimization algorithms for solving (1). However, for ill-conditioned problems, it is often the case that first-order methods return a solution far from the minimizer, $\mathbf{w}^*$, albeit a low objective value

Reference: [Nocedal and Wright, 2006]

Second-order methods

- There is a plethora of first-order optimization algorithms for solving (1). However, for ill-conditioned problems, it is often the case that first-order methods return a solution far from the minimizer, $\mathbf{w}^*$, albeit a low objective value
- On the other hand, most second-order algorithms prove to be more robust to such ill conditioning. This is so since, using the curvature information, second-order methods properly rescale the gradient, such that it is a more appropriate direction to follow

Reference: [Nocedal and Wright, 2006]

Newton's method

Newton's method enjoys fast *local* convergence and is good at recovering the minimizer $\mathbf{w}^*$. In the unconstrained case, it has updates of the form

$$\begin{aligned} \mathbf{H}(\mathbf{w}_t)\mathbf{v} &= \mathbf{g}(\mathbf{w}_t), && (3) \\ \mathbf{w}_{t+1} &= \mathbf{w}_t - \mathbf{v} && (4) \end{aligned}$$

Newton's method

Newton's method enjoys fast *local* convergence and is good at recovering the minimizer $\mathbf{w}^*$. In the unconstrained case, it has updates of the form

$$\begin{aligned} \mathbf{H}(\mathbf{w}_t)\mathbf{v} &= \mathbf{g}(\mathbf{w}_t), && (3) \\ \mathbf{w}_{t+1} &= \mathbf{w}_t - \mathbf{v} && (4) \end{aligned}$$

Issues when n and d are large:

- When n is large, forming the Hessian

$$\mathbf{H}(\mathbf{w}_t) = \sum_{i=1}^{n} \nabla^2 f_i(\mathbf{w}) + \nabla^2 R(\mathbf{w}) := \sum_{i=1}^{n} \mathbf{H}_i(\mathbf{w}) + \mathbf{Q}(\mathbf{w}) \tag{5}$$

  is expensive. The cost is $\mathcal{O}(nd^2)$ in the above example
- When d is large, solving (3) is also expensive: $\mathcal{O}(d^3)$

Remedy

- When n is large, forming the Hessian

$$\mathbf{H}(\mathbf{w}_t) = \sum_{i=1}^{n} \nabla^2 f_i(\mathbf{w}) + \nabla^2 R(\mathbf{w}) := \sum_{i=1}^{n} \mathbf{H}_i(\mathbf{w}) + \mathbf{Q}(\mathbf{w}) \tag{6}$$

  is expensive. The cost is $\mathcal{O}(nd^2)$ in the above example

- **Idea:** Sub-sample only a few terms, say s, from $\{\mathbf{H}_i(\mathbf{w})\}_{i=1}^{n}$, *without* forming them, to form $\tilde{\mathbf{H}}$ so that the cost can be reduced to $\mathcal{O}(sd^2)$

Remedy

- When n is large, forming the Hessian

$$\mathbf{H}(\mathbf{w}_t) = \sum_{i=1}^{n} \nabla^2 f_i(\mathbf{w}) + \nabla^2 R(\mathbf{w}) := \sum_{i=1}^{n} \mathbf{H}_i(\mathbf{w}) + \mathbf{Q}(\mathbf{w}) \tag{6}$$

  is expensive. The cost is $\mathcal{O}(nd^2)$ in the above example

- **Idea:** Sub-sample only a few terms, say s, from $\{\mathbf{H}_i(\mathbf{w})\}_{i=1}^{n}$, *without* forming them, to form $\tilde{\mathbf{H}}$ so that the cost can be reduced to $\mathcal{O}(sd^2)$
- When d is large, solving (3) is also expensive: $\mathcal{O}(d^3)$
- **Idea:** Use an iterative solver such as Conjugate Gradient to solve (3)

Main contributions

- We propose randomized Newton-type algorithms that exploit *non-uniform* sub-sampling of $\{\nabla^2 f_i(\mathbf{w})\}_{i=1}^n$, as well as *inexact updates*, as means to reduce the computational complexity
- Two non-uniform sampling distributions based on *row norm squares* and *leverage scores* are considered in order to capture important terms among $\{\nabla^2 f_i(\mathbf{w})\}_{i=1}^n$
- We show that at each iteration non-uniformly sampling at most $\mathcal{O}(d \log d)$ terms from $\{\nabla^2 f_i(\mathbf{w})\}_{i=1}^n$ is sufficient to achieve a *linear-quadratic convergence rate* in $\mathbf{w}$ when a suitable initial point is provided
- We show that to achieve a locally *problem independent* linear convergence rate, the per-iteration complexities of our algorithm have *lower dependence* on condition numbers compared to [Agarwal et al., 2016, Pilanci and Wainwright, 2015, Roosta-Khorasani and Mahoney, 2016b]
- We empirically demonstrate that our methods are at least *twice* as fast as Newton's methods with ridge logistic regression on several real datasets

Related work

- Newton sketch [Pilanci and Wainwright, 2015] considers a similar class of problems and proposes sketching the Hessian using random sub-Gaussian matrices or randomized orthonormal systems
- Algorithms that employ uniform sub-sampling constitute a popular line of work [Byrd et al., 2011, Erdogdu and Montanari, 2015, Martens, 2010, Vinyals and Povey, 2011]
- Roosta-Khorasani and Mahoney [2016a,b] consider a more general class of problems and, under a variety of conditions, thoroughly study the local and global convergence properties of sub-sampled Newton methods where the gradient and/or the Hessian are uniformly sub-sampled
- Agarwal et al. [2016] proposes a stochastic algorithm (LiSSA) that, for solving the sub-problems, employs some unbiased estimators of the inverse of the Hessian

Roadmap

1 Algorithm description
- Overview of the algorithm
- Non-uniformly sub-sampled Hessian (sampling scheme)
- Inexact updates (solver)

2 Convergence results

3 Empirical results

1 Algorithm description

- Overview of the algorithm
- Non-uniformly sub-sampled Hessian (sampling scheme)
- Inexact updates (solver)

2 Convergence results

3 Empirical results

Sub-sampled Newton methods (SSN)

Algorithm

1. Construct an approximate Hessian $\tilde{\mathbf{H}}(\mathbf{w})$ by *non-uniformly* sub-sampling terms from $\{\mathbf{H}_i(\mathbf{w})\}_{i=1}^n$ *without* forming $\mathbf{H}_i(\mathbf{w})'s$ based on a sampling scheme. The update formula becomes

$$\tilde{\mathbf{H}}(\mathbf{w}_t)\mathbf{v} \quad = \quad \mathbf{g}(\mathbf{w}_t) \tag{7}$$

2. Solve the subproblem (7) using an iterative solver such as CG to return an *approximate* $\mathbf{v}$, denoted by $\tilde{\mathbf{v}}$, and

$$\mathbf{w}_{t+1} \quad = \quad \mathbf{w}_t - \tilde{\mathbf{v}} \tag{8}$$

Complexity

The total complexity can be expressed as

$$T \cdot (t_{grad} + t_{const} + t_{solve}) \tag{9}$$

- Number of total iterations T determined by the convergence rate (sampling scheme and solver)
- t_{grad} is the time it takes to compute the full gradient $\nabla F(\mathbf{w}_t)$ (will not be discussed)
- In each iteration, the time t_{const} it needs to construct $\{p_i\}_{i=1}^n$ and sample s terms (sampling scheme)
- In each iteration, the time t_{solve} it needs to (implicitly) form $\tilde{\mathbf{H}}$ (sampling scheme) and to (inexactly) solve the linear problem (solver)

1 Algorithm description

- Overview of the algorithm
- Non-uniformly sub-sampled Hessian (sampling scheme)
- Inexact updates (solver)

2 Convergence results

3 Empirical results

A simple example

- When $f_i(\mathbf{w}) = \ell(\mathbf{x}_i^T \mathbf{w})$ and $R(\mathbf{w}) = 0$,

$$\mathbf{H}_i(\mathbf{w}) = \nabla^2 f_i(\mathbf{w}) = \ell''(\mathbf{x}_i^T \mathbf{w}) \cdot \mathbf{x}_i \mathbf{x}_i^T \tag{10}$$

A simple example

- When $f_i(\mathbf{w}) = \ell(\mathbf{x}_i^T \mathbf{w})$ and $R(\mathbf{w}) = 0$,

$$\mathbf{H}_i(\mathbf{w}) = \nabla^2 f_i(\mathbf{w}) = \ell''(\mathbf{x}_i^T \mathbf{w}) \cdot \mathbf{x}_i \mathbf{x}_i^T \tag{10}$$

- Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ be a matrix with rows

$$\mathbf{A}_i = (\ell''(\mathbf{x}_i^T \mathbf{w}))^{\frac{1}{2}} \mathbf{x}_i \quad \text{so that} \quad \mathbf{A}_i \mathbf{A}_i^T = \mathbf{H}_i(\mathbf{w}) \tag{11}$$

A simple example

- When $f_i(\mathbf{w}) = \ell(\mathbf{x}_i^T \mathbf{w})$ and $R(\mathbf{w}) = 0$,

$$\mathbf{H}_i(\mathbf{w}) = \nabla^2 f_i(\mathbf{w}) = \ell''(\mathbf{x}_i^T \mathbf{w}) \cdot \mathbf{x}_i \mathbf{x}_i^T \tag{10}$$

- Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ be a matrix with rows

$$\mathbf{A}_i = (\ell''(\mathbf{x}_i^T \mathbf{w}))^{\frac{1}{2}} \mathbf{x}_i \quad \text{so that} \quad \mathbf{A}_i \mathbf{A}_i^T = \mathbf{H}_i(\mathbf{w}) \tag{11}$$

- Forming $\mathbf{A}$ takes $\mathcal{O}(nd)$ time and $\mathbf{A}^T \mathbf{A} = \sum_i \mathbf{H}_i(\mathbf{w}) = \mathbf{H}$ (which needs $\mathcal{O}(nd^2)$ to compute)

A simple example

- When $f_i(\mathbf{w}) = \ell(\mathbf{x}_i^T\mathbf{w})$ and $R(\mathbf{w}) = 0$,

$$\mathbf{H}_i(\mathbf{w}) = \nabla^2 f_i(\mathbf{w}) = \ell''(\mathbf{x}_i^T\mathbf{w}) \cdot \mathbf{x}_i\mathbf{x}_i^T \tag{10}$$

- Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ be a matrix with rows

$$\mathbf{A}_i = (\ell''(\mathbf{x}_i^T\mathbf{w}))^{\frac{1}{2}}\mathbf{x}_i \quad \text{so that} \quad \mathbf{A}_i\mathbf{A}_i^T = \mathbf{H}_i(\mathbf{w}) \tag{11}$$

- Forming $\mathbf{A}$ takes $\mathcal{O}(nd)$ time and $\mathbf{A}^T\mathbf{A} = \sum_i \mathbf{H}_i(\mathbf{w}) = \mathbf{H}$ (which needs $\mathcal{O}(nd^2)$ to compute)
- Consider sub-sampling rows from $\mathbf{A}$ such that

$$\mathbf{H}(\mathbf{w}) = \mathbf{A}^T\mathbf{A} \approx \mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A} = \tilde{\mathbf{H}}(\mathbf{w}) \tag{12}$$

The running time is reduced to $\mathcal{O}(sd^2)$ from $\mathcal{O}(nd^2)$

General case

- Assume each $\mathbf{H}_i(\mathbf{w})$ has a low-rank decomposition readily accessible: $\mathbf{H}_i(\mathbf{w}) = \mathbf{A}_i \mathbf{A}_i^T$ where $\mathbf{A}_i \in \mathbb{R}^{d \times k_i}$
- Further assume that $k_i = k = \mathcal{O}(1)$ ($k_i = 1$ in the above example)
- Denote $\mathbf{Q} = \nabla^2 R(\mathbf{w})$

General case

- Assume each $\mathbf{H}_i(\mathbf{w})$ has a low-rank decomposition readily accessible: $\mathbf{H}_i(\mathbf{w}) = \mathbf{A}_i \mathbf{A}_i^T$ where $\mathbf{A}_i \in \mathbb{R}^{d \times k_i}$
- Further assume that $k_i = k = \mathcal{O}(1)$ ($k_i = 1$ in the above example)
- Denote $\mathbf{Q} = \nabla^2 R(\mathbf{w})$
- Then

$$\nabla^2 f(\mathbf{w}) = \mathbf{H}(\mathbf{w}) = \sum_{i=1}^{n} \mathbf{H}_i(\mathbf{w}) + \mathbf{Q} = \mathbf{A}^T \mathbf{A} + \mathbf{Q}, \tag{13}$$

where $\mathbf{A} = \begin{pmatrix} \mathbf{A}_1^T & \cdots & \mathbf{A}_n^T \end{pmatrix}^T \in \mathbb{R}^{kn \times d}$

General case

- Assume each $\mathbf{H}_i(\mathbf{w})$ has a low-rank decomposition readily accessible: $\mathbf{H}_i(\mathbf{w}) = \mathbf{A}_i \mathbf{A}_i^T$ where $\mathbf{A}_i \in \mathbb{R}^{d \times k_i}$
- Further assume that $k_i = k = \mathcal{O}(1)$ ($k_i = 1$ in the above example)
- Denote $\mathbf{Q} = \nabla^2 R(\mathbf{w})$
- Then

$$\nabla^2 f(\mathbf{w}) = \mathbf{H}(\mathbf{w}) = \sum_{i=1}^{n} \mathbf{H}_i(\mathbf{w}) + \mathbf{Q} = \mathbf{A}^T \mathbf{A} + \mathbf{Q}, \tag{13}$$

  where $\mathbf{A} = \begin{pmatrix} \mathbf{A}_1^T & \cdots & \mathbf{A}_n^T \end{pmatrix}^T \in \mathbb{R}^{kn \times d}$

- The task becomes sub-sampling *blocks* from $\mathbf{A}$ such that

$$\mathbf{H} = \mathbf{A}^T \mathbf{A} + \mathbf{Q} \approx \mathbf{A}^T \mathbf{S}^T \mathbf{S} \mathbf{A} + \mathbf{Q} = \tilde{\mathbf{H}} \tag{14}$$

General case

- Assume each $\mathbf{H}_i(\mathbf{w})$ has a low-rank decomposition readily accessible: $\mathbf{H}_i(\mathbf{w}) = \mathbf{A}_i\mathbf{A}_i^T$ where $\mathbf{A}_i \in \mathbb{R}^{d\times k_i}$
- Further assume that $k_i = k = \mathcal{O}(1)$ ($k_i = 1$ in the above example)
- Denote $\mathbf{Q} = \nabla^2 R(\mathbf{w})$
- Then

$$\nabla^2 f(\mathbf{w}) = \mathbf{H}(\mathbf{w}) = \sum_{i=1}^{n} \mathbf{H}_i(\mathbf{w}) + \mathbf{Q} = \mathbf{A}^T\mathbf{A} + \mathbf{Q}, \tag{13}$$

  where $\mathbf{A} = \begin{pmatrix}\mathbf{A}_1^T & \cdots & \mathbf{A}_n^T\end{pmatrix}^T \in \mathbb{R}^{kn\times d}$
- The task becomes sub-sampling *blocks* from $\mathbf{A}$ such that

$$\mathbf{H} = \mathbf{A}^T\mathbf{A} + \mathbf{Q} \approx \mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A} + \mathbf{Q} = \tilde{\mathbf{H}} \tag{14}$$

- This is similar to the matrix approximation problem:

$$\mathbf{A}^T\mathbf{A} \approx \mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A} \tag{15}$$

Sufficient conditions for matrix approximation

By $\mathbf{H} = \mathbf{A}^T\mathbf{A} + \mathbf{Q} \approx \mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A} + \mathbf{Q} = \tilde{\mathbf{H}}$, we mean one of the followings

- ℓ_2 norm guarantee:

$$\|(\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A} + \mathbf{Q}) - (\mathbf{A}^T\mathbf{A} + \mathbf{Q})\| \leq \epsilon\|\mathbf{A}^T\mathbf{A} + \mathbf{Q}\| \qquad \textbf{(C1)}$$

- Spectral guarantee:

$$-\epsilon(\mathbf{A}^T\mathbf{A} + \mathbf{Q}) \preceq (\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A} + \mathbf{Q}) - (\mathbf{A}^T\mathbf{A} + \mathbf{Q}) \preceq \epsilon(\mathbf{A}^T\mathbf{A} + \mathbf{Q}) \qquad \textbf{(C2)}$$

Sufficient conditions for matrix approximation

By $\mathbf{H} = \mathbf{A}^T\mathbf{A} + \mathbf{Q} \approx \mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A} + \mathbf{Q} = \tilde{\mathbf{H}}$, we mean one of the followings

- ℓ_2 norm guarantee:

$$\|(\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A} + \mathbf{Q}) - (\mathbf{A}^T\mathbf{A} + \mathbf{Q})\| \leq \epsilon\|\mathbf{A}^T\mathbf{A} + \mathbf{Q}\| \qquad \textbf{(C1)}$$

- Spectral guarantee:

$$-\epsilon(\mathbf{A}^T\mathbf{A} + \mathbf{Q}) \preceq (\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A} + \mathbf{Q}) - (\mathbf{A}^T\mathbf{A} + \mathbf{Q}) \preceq \epsilon(\mathbf{A}^T\mathbf{A} + \mathbf{Q}) \qquad \textbf{(C2)}$$

As we can see, (**C2**) is stronger than (**C1**), and we will show that (**C2**) leads to a better convergence rate

Sufficient conditions for matrix approximation

By $\mathbf{H} = \mathbf{A}^T\mathbf{A} + \mathbf{Q} \approx \mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A} + \mathbf{Q} = \tilde{\mathbf{H}}$, we mean one of the followings

- ℓ_2 norm guarantee:

$$\|(\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A} + \mathbf{Q}) - (\mathbf{A}^T\mathbf{A} + \mathbf{Q})\| \leq \epsilon\|\mathbf{A}^T\mathbf{A} + \mathbf{Q}\| \qquad \textbf{(C1)}$$

- Spectral guarantee:

$$-\epsilon(\mathbf{A}^T\mathbf{A} + \mathbf{Q}) \preceq (\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A} + \mathbf{Q}) - (\mathbf{A}^T\mathbf{A} + \mathbf{Q}) \preceq \epsilon(\mathbf{A}^T\mathbf{A} + \mathbf{Q}) \qquad \textbf{(C2)}$$

As we can see, (**C2**) is stronger than (**C1**), and we will show that (**C2**) leads to a better convergence rate

Two non-uniform sampling techniques in randomized linear algebra (RLA) are considered: *leverage scores sampling* (achieves (**C2**)) and *row norm squares sampling* (achieves (**C1**))

Standard leverage scores sampling

Definition (Leverage scores)

Given $\mathbf{A} \in \mathbb{R}^{n \times d}$, then for $i = 1, \ldots, n$, the i-th leverage scores of $\mathbf{A}$ is defined as

$$\tau_i(\mathbf{A}) = \mathbf{a}_i^T (\mathbf{A}^T \mathbf{A})^\dagger \mathbf{a}_i \tag{16}$$

Standard leverage scores sampling

Definition (Leverage scores)

Given $\mathbf{A} \in \mathbb{R}^{n \times d}$, then for $i = 1, \ldots, n$, the i-th leverage scores of $\mathbf{A}$ is defined as

$$\tau_i(\mathbf{A}) = \mathbf{a}_i^T (\mathbf{A}^T \mathbf{A})^\dagger \mathbf{a}_i \tag{16}$$

Theorem ([Mahoney, 2011])

Given $\mathbf{A}$, if $\mathcal{O}(d \log d / \epsilon^2)$ rows are sampled according to leverage scores, then

$$-\epsilon \mathbf{A}^T \mathbf{A} \preceq \mathbf{A}^T \mathbf{S}^T \mathbf{S} \mathbf{A} - \mathbf{A}^T \mathbf{A} \preceq \epsilon \mathbf{A}^T \mathbf{A} \tag{17}$$

Standard leverage scores sampling

Definition (Leverage scores)

Given $\mathbf{A} \in \mathbb{R}^{n \times d}$, then for $i = 1, \ldots, n$, the i-th leverage scores of $\mathbf{A}$ is defined as

$$\tau_i(\mathbf{A}) = \mathbf{a}_i^T (\mathbf{A}^T \mathbf{A})^{\dagger} \mathbf{a}_i \tag{16}$$

Theorem ([Mahoney, 2011])

Given $\mathbf{A}$, if $\mathcal{O}(d \log d / \epsilon^2)$ rows are sampled according to leverage scores, then

$$-\epsilon \mathbf{A}^T \mathbf{A} \preceq \mathbf{A}^T \mathbf{S}^T \mathbf{S} \mathbf{A} - \mathbf{A}^T \mathbf{A} \preceq \epsilon \mathbf{A}^T \mathbf{A} \tag{17}$$

Recall that, there are two main differences between (17) and (**C2**)

- Blocks of $\mathbf{A}$ are being sampled, instead of rows
- An additional matrix $\mathbf{Q}$ is involved in the target matrix $\mathbf{A}^T \mathbf{A} + \mathbf{Q}$

Remedy

- For the first difference, inspired by the work [de Carli Silva et al., 2011], define the leverage score of a “block” by summing up the leverage scores in that block

Remedy

- For the first difference, inspired by the work [de Carli Silva et al., 2011], define the leverage score of a “block” by summing up the leverage scores in that block
- For the second difference, a naive idea is construct $\mathbf{S}$ based on information of $\mathbf{A}$ only, ignoring $\mathbf{Q}$

Remedy

- For the first difference, inspired by the work [de Carli Silva et al., 2011], define the leverage score of a "block" by summing up the leverage scores in that block
- For the second difference, a naive idea is construct $\mathbf{S}$ based on information of $\mathbf{A}$ only, ignoring $\mathbf{Q}$
- However, we can do something better (minimize sampling size)

Remedy

- For the first difference, inspired by the work [de Carli Silva et al., 2011], define the leverage score of a "block" by summing up the leverage scores in that block
- For the second difference, a naive idea is construct $\mathbf{S}$ based on information of $\mathbf{A}$ only, ignoring $\mathbf{Q}$
- However, we can do something better (minimize sampling size)
- Inspired by the recently proposed ridge leverage scores by El Alaoui and Mahoney [2014], Cohen et al. [2015], consider leverage scores of a matrix that concatenates $\mathbf{A}$ and $\mathbf{Q}^{\frac{1}{2}}$ since essentially we are essentially approximating

$$\mathbf{A}^T\mathbf{A} + \mathbf{Q} = \mathbf{B}^T\mathbf{B}, \tag{18}$$

where $\mathbf{B} = \begin{pmatrix} \mathbf{A} \\ \mathbf{Q}^{\frac{1}{2}} \end{pmatrix}$

Block partial leverage scores

Definition (Block partial leverage scores)

Given a matrix $\mathbf{A} \in \mathbb{R}^{kn \times d}$ with n blocks and a matrix $\mathbf{Q} \in \mathbb{R}^{d \times d}$ satisfying $\mathbf{Q} \succeq \mathbf{0}$, let $\{\tau_i\}_{i=1}^{kn+d}$ be the leverage scores of the matrix $\begin{pmatrix} \mathbf{A} \\ \mathbf{Q}^{\frac{1}{2}} \end{pmatrix}$. Define the block partial leverage score for the i-th block as

$$\tau_i^{\mathbf{Q}}(\mathbf{A}) = \sum_{j=k(i-1)+1}^{ki} \tau_j$$

Sampling size

Theorem (Xu, Y, Roosta-Khorasani, Ré and Mahoney [2016])

Given $\mathbf{A} \in \mathbb{R}^{N \times d}$ with n blocks, $\mathbf{Q} \in \mathbb{R}^{d \times d}$ satisfying $\mathbf{Q} \succeq \mathbf{0}$ and $\epsilon \in (0,1)$, if $\mathbf{S}$ is constructed based on the block partial leverage scores $\tau_i^{\mathbf{Q}}(\mathbf{A})$ and

$$s \geq 4 \left(\sum_{i=1}^{n} \tau_i^{\mathbf{Q}}(\mathbf{A}) \right) \cdot \log \left(\frac{4d}{\delta} \right) \cdot \frac{1}{\epsilon^2}, \tag{19}$$

with probability at least $1-\delta$, **(C2)** *is satisfied, i.e.,*

$$-\epsilon(\mathbf{A}^T\mathbf{A} + \mathbf{Q}) \preceq (\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A} + \mathbf{Q}) - (\mathbf{A}^T\mathbf{A} + \mathbf{Q}) \preceq \epsilon(\mathbf{A}^T\mathbf{A} + \mathbf{Q}) \tag{20}$$

Here, $\sum_{i=1}^{n} \tau_i^{\mathbf{Q}}(\mathbf{A}) \leq d$ always holds. In some cases, it can be much smaller than d

Construction time

- Since the block partial leverage scores are defined as the standard leverage scores of some matrix, we can make use of the fast approximation algorithm for standard leverage scores [Drineas et al., 2012]
- The high-level idea is

$$\ell_i = \|\mathbf{e}_i \mathbf{A}\mathbf{A}^\dagger\| \approx \|\mathbf{e}_i \mathbf{A}(\mathbf{\Phi}_1 \mathbf{A})^\dagger\| \approx \|\mathbf{e}_i \mathbf{A}(\mathbf{\Phi}_1 \mathbf{A})^\dagger \mathbf{\Phi}_2\| \tag{21}$$

- Here we use the sparse subspace embedding [Clarkson and Woodruff, 2013] as $\mathbf{\Phi}_1$ and Gaussian transform as $\mathbf{\Phi}_2$

Theorem

It takes $t_{const} = \mathcal{O}(\mathrm{nnz}(\mathbf{A}) \log n)$ *time to construct a set of* β*-approximate leverage scores* $\{\hat{\tau}_i^{\mathbf{Q}}(\mathbf{A})\}_{i=1}^n$ *such that with high probability,*

$$\tau_i^{\mathbf{Q}}(\mathbf{A}) \leq \hat{\tau}_i^{\mathbf{Q}}(\mathbf{A}) \leq \beta \cdot \tau_i^{\mathbf{Q}}(\mathbf{A}) \tag{22}$$

where $\{\tau_i\}_{i=1}^n$ *are the block partial leverage scores of* $\mathbf{A}$ *given* $\mathbf{Q}$

Block norm squares sampling

- Another sampling technique we consider here is based on row norm squares sampling
- Since we are sampling blocks, we sample based on the “magnitude” of blocks, i.e., $\|\mathbf{A}_i\|_F^2$
- We don’t know how to incorporate $\mathbf{Q}$ into the construction of the distribution in this case

Sampling size

Theorem ([Holodnak and Ipsen, 2015])

Given $\mathbf{A}$ *with* n *blocks,* $\mathbf{Q} \succeq \mathbf{0}$ *and* $\epsilon \in (0, 1)$*, for* $i = 1, \ldots, n$*, let* $r_i = \|\mathbf{A}_i\|_F^2$*. If* $\mathbf{S}$ *is constructed based on* $\{r_i\}_{i=1}^n$ *and*

$$s \geq 4 \cdot \mathbf{sr}(\mathbf{A}) \cdot \log \frac{\min\{4\mathbf{sr}(\mathbf{A}), d\}}{\delta} \cdot \frac{1}{\epsilon^2}, \tag{23}$$

with probability at least $1 - \delta$, **(C1)** *is satisfied, i.e.,*

$$\|(\mathbf{A}^T\mathbf{S}^T\mathbf{S}\mathbf{A} + \mathbf{Q}) - (\mathbf{A}^T\mathbf{A} + \mathbf{Q})\| \leq \epsilon\|\mathbf{A}^T\mathbf{A} + \mathbf{Q}\| \tag{24}$$

Here, $\mathbf{sr}(\mathbf{A})$ denotes the stable rank which satisfies $\mathbf{sr}(\mathbf{A}) \leq d$

Recap: Sub-sampled Newton methods (SSN)

Algorithm

1. Construct an approximate Hessian $\tilde{\mathbf{H}}(\mathbf{w})$ by *non-uniformly* sub-sampling terms from $\{\mathbf{H}_i(\mathbf{w})\}_{i=1}^n$ *without* forming $\mathbf{H}_i(\mathbf{w})'s$ based on a sampling scheme. The update formula becomes

$$\tilde{\mathbf{H}}(\mathbf{w}_t)\mathbf{v} \quad = \quad \mathbf{g}(\mathbf{w}_t) \tag{25}$$

2. Solve the subproblem (25) using an iterative solver such as CG to return an *approximate* $\mathbf{v}$, denoted by $\tilde{\mathbf{v}}$, and

$$\mathbf{w}_{t+1} \quad = \quad \mathbf{w}_t - \tilde{\mathbf{v}} \tag{26}$$

1 Algorithm description

- Overview of the algorithm
- Non-uniformly sub-sampled Hessian (sampling scheme)
- Inexact updates (solver)

2 Convergence results

3 Empirical results

Sufficient condition for the solver

- Want to solve

$$\tilde{\mathbf{H}}_t \mathbf{v} = -\nabla F(\mathbf{w}_t) \tag{27}$$

- Require the solver to return an approximate solution $\mathbf{v}$ such that

$$\|\mathbf{v} - \mathbf{v}^*\| \leq \epsilon_0 \|\mathbf{v}^*\|, \tag{28}$$

where $\mathbf{v}^*$ is the optimal solution to (27)

Solvers

SOLVER	TIME	ϵ_0	REFERENCE
direct	$\mathcal{O}(sd^2)$	0	[Golub and Van Loan, 2012]
CG	$\mathcal{O}(sd\sqrt{\tilde{\kappa}_t}\log(1/\epsilon))$	$\sqrt{\tilde{\kappa}_t}\epsilon$	[Golub and Van Loan, 2012]
GD	$\mathcal{O}(sd\tilde{\kappa}_t\log(1/\epsilon))$	ϵ	[Nesterov, 2004, Theorem 2.1.15]
ACDM	$\mathcal{O}(s\mathbf{sr}(\mathbf{SA})\sqrt{\tilde{\kappa}_t}\log(1/\epsilon))$	$\sqrt{\tilde{\kappa}_t}\epsilon$	[Lee and Sidford, 2013]

Table: Comparison of different solvers. Here $\tilde{\kappa}_t = \lambda_{\max}(\tilde{\mathbf{H}}_t)/\lambda_{\min}(\tilde{\mathbf{H}}_t)$

Solvers

SOLVER	TIME	ϵ_0	REFERENCE
direct	$\mathcal{O}(sd^2)$	0	[Golub and Van Loan, 2012]
CG	$\mathcal{O}(sd\sqrt{\tilde{\kappa}_t}\log(1/\epsilon))$	$\sqrt{\tilde{\kappa}_t}\epsilon$	[Golub and Van Loan, 2012]
GD	$\mathcal{O}(sd\tilde{\kappa}_t\log(1/\epsilon))$	ϵ	[Nesterov, 2004, Theorem 2.1.15]
ACDM	$\mathcal{O}(s\mathbf{sr}(\mathbf{SA})\sqrt{\tilde{\kappa}_t}\log(1/\epsilon))$	$\sqrt{\tilde{\kappa}_t}\epsilon$	[Lee and Sidford, 2013]

Table: Comparison of different solvers. Here $\tilde{\kappa}_t = \lambda_{\max}(\tilde{\mathbf{H}}_t)/\lambda_{\min}(\tilde{\mathbf{H}}_t)$

- Can actually solve the subproblem $\tilde{\mathbf{H}}_t\mathbf{v} = -\nabla F(\mathbf{w}_t)$ in a "Hessian-free" manner (without forming $\tilde{\mathbf{H}}_t$ which takes $\mathcal{O}(sd^2)$ time)
- In CG, only $\tilde{\mathbf{H}}_t\mathbf{w}$ needs to be evaluated
- Recall that, $\tilde{\mathbf{H}}_t = (\mathbf{SA})^T(\mathbf{SA}) + \mathbf{Q}$ where $\mathbf{SA} \in \mathbb{R}^d$ can be easily formed without forming $\tilde{\mathbf{H}}_t$
- Equivalent to

$$\tilde{\mathbf{H}}_t\mathbf{w} = (\mathbf{SA})^T\left[(\mathbf{SA})\mathbf{w}\right] + \mathbf{Q}\mathbf{w} \tag{29}$$

- Each evaluation takes only $\mathcal{O}(sd)$ time

Recap: Sub-sampled Newton methods (SSN)

Algorithm

1. Construct an approximate Hessian $\tilde{\mathbf{H}}(\mathbf{w})$ by *non-uniformly* sub-sampling terms from $\{\mathbf{H}_i(\mathbf{w})\}_{i=1}^n$ *without* forming $\mathbf{H}_i(\mathbf{w})'s$ based on a sampling scheme. The update formula becomes

$$\tilde{\mathbf{H}}(\mathbf{w}_t)\mathbf{v} = \mathbf{g}(\mathbf{w}_t) \tag{30}$$

2. Solve the subproblem (30) using an iterative solver such as CG to return an *approximate* $\mathbf{v}$, denoted by $\tilde{\mathbf{v}}$, and

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \tilde{\mathbf{v}} \tag{31}$$

1 Algorithm description

2 Convergence results

3 Empirical results

Assumptions

Assumption (Lipschitz continuity)

$F(\mathbf{w})$ is convex and twice differentiable. The Hessian is L-Lipschitz continuous, i.e.

$$\|\nabla^2 F(\mathbf{u}) - \nabla^2 F(\mathbf{v})\| \leq L\|\mathbf{u} - \mathbf{v}\|, \quad \forall \mathbf{u}, \mathbf{v} \in \mathcal{C}$$

Assumption (Local regularity)

$F(\mathbf{x})$ is locally strongly convex and smooth, i.e.,

$$\mu = \lambda_{\min}^{\mathcal{K}}(\nabla^2 F(\mathbf{w}^*)) > 0, \quad \nu^{\mathcal{K}} = \lambda_{\max}(\nabla^2 F(\mathbf{w}^*)) < \infty$$

Here we define the local condition number of the problem as $\kappa := \nu/\mu$

Convergence results of SSN (exact updates)

Theorem (Xu, **Y**, Roosta-Khorasani, Ré and Mahoney [2016])

If the initial point $\mathbf{w}_0$ *satisfies* $\|\mathbf{w}_0 - \mathbf{w}^*\| \le \frac{\mu}{4L}$ *and condition* (**C1**) *or* (**C2**) *is met, then the solution error satisfies the following recursion*

$$\|\mathbf{w}_{t+1} - \mathbf{w}^*\| \le C_q \cdot \|\mathbf{w}_t - \mathbf{w}^*\|^2 + C_l \cdot \|\mathbf{w}_t - \mathbf{w}^*\|, \tag{32}$$

where C_q *and* C_l *are specified below. Given any* ϵ *small enough,*

- *If the approximate Hessian* $\tilde{\mathbf{H}}_t$ *satisfies* (**C1**), *then in* (32)

$$C_q = \frac{2L}{(1-2\epsilon\kappa)\mu}, \quad C_l = \frac{4\epsilon\kappa}{1-2\epsilon\kappa} \tag{33}$$

- *If the approximate Hessian* $\tilde{\mathbf{H}}_t$ *satisfies* (**C2**), *then in* (32)

$$C_q = \frac{2L}{(1-\epsilon)\mu}, \quad C_l = \frac{3\epsilon}{1-\epsilon}\sqrt{\kappa} \tag{34}$$

Convergence results of SSN (inexact updates)

Theorem (Xu, **Y**, Roosta-Khorasani, Ré and Mahoney [2016])

If an inexact solution is returned when solving the subproblem satisfying

$$\|\mathbf{w}_{t+1} - \mathbf{w}^*_{t+1}\| \le \epsilon_0 \cdot \|\mathbf{w}_t - \mathbf{w}^*_{t+1}\|, \tag{35}$$

then

$$\|\mathbf{w}_{t+1} - \mathbf{w}^*\| \le (1+\epsilon_0)C_q \cdot \|\mathbf{w}_t - \mathbf{w}^*\|^2 + (\epsilon_0 + (1+\epsilon_0)C_l) \cdot \|\mathbf{w}_t - \mathbf{w}^*\| \tag{36}$$

Non-uniform sampling vs. uniform sampling

According to the above, our methods can achieve the following linear-quadratic convergence rate

$$\|\mathbf{w}_{t+1} - \mathbf{w}^*\| \leq C_q \cdot \|\mathbf{w}_t - \mathbf{w}^*\|^2 + C_l \cdot \|\mathbf{w}_t - \mathbf{w}^*\|, \qquad (37)$$

where C_q and C_l are specified below

NAME	t_{const}	SAMPLING SIZE s	C_q	C_l	
SSN (leverage scores)	$\mathcal{O}(\mathrm{nnz}(\mathbf{A}) \log n)$	$\tilde{\mathcal{O}}(\sum_i \tau_i(\mathbf{A})/\epsilon^2)$	$\frac{\tilde{\kappa}}{1-\epsilon}$	$\frac{\epsilon\sqrt{\kappa}}{1-\epsilon}$	**(C2)**
SSN (norm squares)	$\mathcal{O}(\mathrm{nnz}(\mathbf{A}))$	$\tilde{\mathcal{O}}(\mathbf{sr}(\mathbf{A})/\epsilon^2)$	$\frac{\tilde{\kappa}}{1-\epsilon\kappa}$	$\frac{\epsilon\kappa}{1-\epsilon\kappa}$	**(C1)**
SSN (uniform)	$\mathcal{O}(1)$	$\tilde{\mathcal{O}}\left(n \frac{\max_i \|\mathbf{A}_i\|^2}{\|\mathbf{A}\|^2} / \epsilon^2\right)$	$\frac{\tilde{\kappa}}{1-\epsilon\kappa}$	$\frac{\epsilon\kappa}{1-\epsilon\kappa}$	**(C1)**

Table: Convergence rate comparison. Here κ is the problem condition number; $\tilde{\kappa}$ depends on the problem only; $\mathbf{sr}(\mathbf{A})$ is the stable rank satisfying $\mathbf{sr}(\mathbf{A}) \leq d$; $\sum_i \tau_i(\mathbf{A})$ is the sum of block partial leverage scores satisfying $\sum_i \tau_i(\mathbf{A}) \leq d$

Complexity

When a local *problem independent* linear convergence rate, i.e.,

$$\|\mathbf{w}_{t+1} - \mathbf{w}^*\| \leq \rho \cdot \|\mathbf{w}_t - \mathbf{w}^*\| \tag{38}$$

for some fixed $0 < \rho < 1$, is desired, our approach has the following complexity

Complexity

When a local *problem independent* linear convergence rate, i.e.,

$$\|\mathbf{w}_{t+1} - \mathbf{w}^*\| \leq \rho \cdot \|\mathbf{w}_t - \mathbf{w}^*\| \tag{38}$$

for some fixed $0 < \rho < 1$, is desired, our approach has the following complexity

METHOD	COMPLEXITY PER ITERATION	REFERENCE
Newton-CG method	$\tilde{\mathcal{O}}(\text{nnz}(\mathbf{A})\sqrt{\kappa})$	[Nocedal and Wright, 2006]
SSN (leverage scores)	$\tilde{\mathcal{O}}(\text{nnz}(\mathbf{A}) \log n + (\sum_i \tau_i(A))d\kappa^{3/2})$	**This work**
SSN (row norm squares)	$\tilde{\mathcal{O}}(\text{nnz}(\mathbf{A}) + \mathbf{sr}(\mathbf{A})d\kappa^{5/2})$	**This work**
Newton Sketch (SRHT)	$\tilde{\mathcal{O}}(nd(\log n)^4 + d^2(\log n)^4\kappa^{3/2})$	[Pilanci and Wainwright, 2015]
SSN (uniform)	$\tilde{\mathcal{O}}(\text{nnz}(\mathbf{A}) + d\hat{\kappa}\kappa^{3/2})$	[Roosta-Khorasani and Mahoney, 2016b
LiSSA	$\tilde{\mathcal{O}}(\text{nnz}(\mathbf{A}) + d\hat{\kappa}\bar{\kappa}^2)$	[Agarwal et al., 2016]

Table: Complexity per iteration of different methods to obtain a problem independent local linear convergence rate; $\mathbf{sr}(\mathbf{A})$ is the stable rank satisfying $\mathbf{sr}(\mathbf{A}) \leq d$; $\sum_i \tau_i(\mathbf{A})$ is the sum of block partial leverage scores satisfying $\sum_i \tau_i(\mathbf{A}) \leq d$

- $\kappa(\mathbf{w}) = \frac{\lambda_{\max}(\sum_{i=1}^n \mathbf{H}_i(\mathbf{w}))}{\lambda_{\min}(\sum_{i=1}^n \mathbf{H}_i(\mathbf{w}))}, \hat{\kappa}(\mathbf{w}) = n \cdot \frac{\max_i \lambda_{\max}(\mathbf{H}_i(\mathbf{w}))}{\lambda_{\min}(\sum_{i=1}^n \mathbf{H}_i(\mathbf{w}))}, \bar{\kappa}(\mathbf{w}) = \frac{\max_i \lambda_{\max}(\mathbf{H}_i(\mathbf{w}))}{\min_i \lambda_{\min}(\mathbf{H}_i(\mathbf{w}))}$
- Dependence on the condition number is smaller using SSN (leverage scores), e.g., $\kappa^{3/2}$ versus $\hat{\kappa}\kappa^{3/2}$
- $\hat{\kappa}$ can be a factor of n higher than κ

1 Algorithm description

2 Convergence results

3 Empirical results

Ridge logistic regression

- Assume $\mathbf{X} \in \mathbb{R}^{n \times d}, \mathbf{Y} \in \{\pm 1\}^n$ are the data matrix and response vector
- Want to solve

$$\min_{\mathbf{w} \in \mathbb{R}^d} \sum_{i=1}^{n} \psi(\mathbf{x}_i^T \mathbf{w}, y_i) + \lambda \|\mathbf{w}\|_2^2, \tag{39}$$

  where $\psi(u, y) = \log(1 + \exp(-uy))$

- In this case, the Hessian is

$$\mathbf{H}(\mathbf{w}) = \sum_{i=1}^{n} \psi''(\mathbf{x}_i^T \mathbf{w}, y_i) \mathbf{x}_i \mathbf{x}_i^T + \lambda \mathbf{I} := \mathbf{X}^T \mathbf{D}^2(\mathbf{w}) \mathbf{X} + \lambda \mathbf{I}, \tag{40}$$

  where $\mathbf{x}_i$ is i-th column of $\mathbf{X}^T$ and $\mathbf{D}(\mathbf{w})$ is a diagonal matrix with the diagonal $[\mathbf{D}(\mathbf{w})]_{ii} = \sqrt{\psi''(\mathbf{x}_i^T \mathbf{w}, y_i)}$

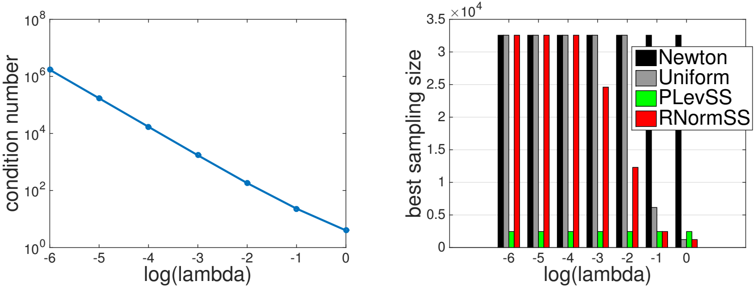
- The matrix $\mathbf{A}$ can be written as $\mathbf{A} = \mathbf{D}(\mathbf{w})\mathbf{X} \in \mathbb{R}^{n \times d}$ where $\mathbf{A}_i = [\mathbf{D}(\mathbf{w})]_{ii} \mathbf{x}_i^T$

Datasets

DATASET	CT slices	Forest	Adult	Buzz
n	53,500	581,012	32,561	59,535
d	385	55	123	78
κ	368	221	182	37
$\hat{\kappa}$	47,078	322,370	69,359	384,580

Table: Datasets used in ridge logistic regression. In the above, κ and $\hat{\kappa}$ are the local condition numbers of ridge logistic regression problem with $\lambda = 0.01$ defined previously

Condition number

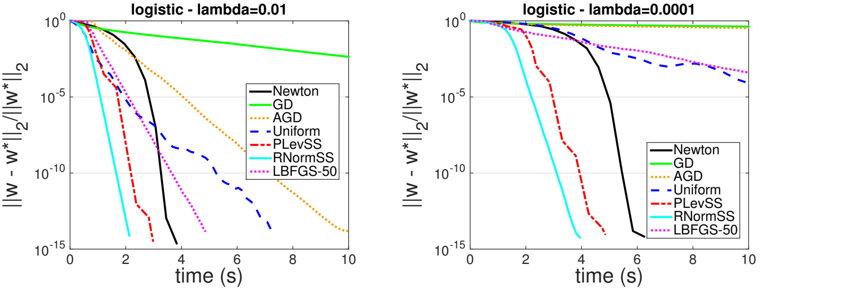


(a) condition number

(b) sampling size

Figure: Ridge logistic regression on `Adult` with different λ's: (a) local condition number κ, (b) sample size for different SSN methods giving the best overall running time

First-order vs. second-order methods

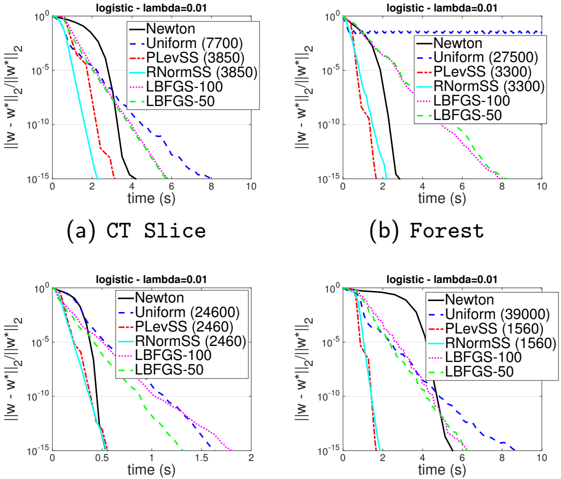


(a) better conditioned

(b) worse conditioned

Figure: Iterate relative error vs. time(s) for a ridge logistic regression problem with two choices of regularization parameter λ on a real dataset `CT Slice`

Time-accuracy tradeoffs



(a) CT Slice

(b) Forest

(c) Adult

(d) Buzz

Figure: Iterate relative solution error vs. time(s) for various second-order methods. The values in brackets denote the sample size used for each method

Conclusion

- We propose non-uniformly sub-sampled Newton methods with inexact update for a class of constrained problems
- Two non-uniform sampling distributions based on block norm squares and a new notion, block partial leverage scores, are considered
- We show that at each iteration non-uniformly sampling at most $\mathcal{O}(d \log d)$ terms is sufficient to achieve a linear-quadratic convergence rate
- We show that our algorithms have a better dependence on the condition number and enjoy a lower per-iteration complexity, compared to other similar existing methods
- We numerically verify the advantages of our algorithms on several real datasets