Randomized Second Order Method

Kiyeon Jeon

University of Texas at Austin

kiyeonj@utexas.edu

December 1, 2016

Outline

Machine Learning Model

- Given dataset $X \in \mathbb{R}^{n \times d}$ and label $y \in \mathcal{C}^n$,

Machine Learning Model

- Given dataset $X \in \mathbb{R}^{n \times d}$ and label $y \in \mathcal{C}^n$,
- $\min_{\beta \in \mathbb{R}^d} f(\beta) := \sum_{i=1}^{n} f_i(\beta)$

Machine Learning Model

- Given dataset $X \in \mathbb{R}^{n \times d}$ and label $y \in \mathcal{C}^n$,
- $\min_{\beta \in \mathbb{R}^d} f(\beta) := \sum_{i=1}^{n} f_i(\beta)$
- $f, f_i : \mathbb{R}^d \to \mathbb{R}$

Machine Learning Model

- Given dataset $X \in \mathbb{R}^{n \times d}$ and label $y \in \mathcal{C}^n$,
- $\min\limits_{\beta \in \mathbb{R}^d} f(\beta) := \sum_{i=1}^{n} f_i(\beta)$
- $f, f_i : \mathbb{R}^d \to \mathbb{R}$
- $f_i(\beta) = \psi(x_i^T \beta, y_i) := g_i(x_i^T \beta)$

Machine Learning Model

- Given dataset $X \in \mathbb{R}^{n \times d}$ and label $y \in \mathcal{C}^n$,
- $\min_{\beta \in \mathbb{R}^d} f(\beta) := \sum_{i=1}^n f_i(\beta)$
- $f, f_i : \mathbb{R}^d \to \mathbb{R}$
- $f_i(\beta) = \psi(x_i^T \beta, y_i) := g_i(x_i^T \beta)$
- $n \gg d \gg 1$

Machine Learning Model

- Given dataset $X \in \mathbb{R}^{n \times d}$ and label $y \in \mathcal{C}^n$,
- $\min\limits_{\beta \in \mathbb{R}^d} f(\beta) := \sum_{i=1}^n f_i(\beta)$
- $f, f_i : \mathbb{R}^d \to \mathbb{R}$
- $f_i(\beta) = \psi(x_i^T \beta, y_i) := g_i(x_i^T \beta)$
- $n \gg d \gg 1$
- Application : logistic regression, support vector machine, neural networks and graphical model

Machine Learning Model

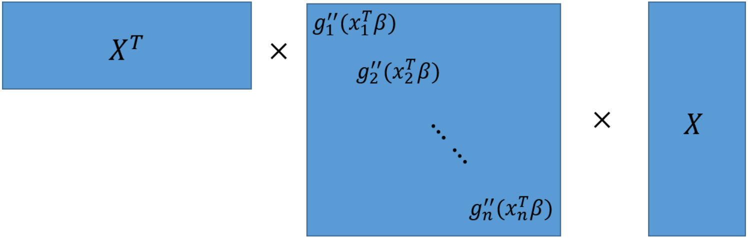
- $f(\beta) = \sum_{i=1}^{n} g_i(x_i^T \beta)$

Machine Learning Model

- $f(\beta) = \sum_{i=1}^{n} g_i(x_i^T \beta)$
- $\nabla^2 f(\beta) = X^T \mathrm{diag}\{g_i''(x_i^T \beta)\}_{i=1}^{n} X$

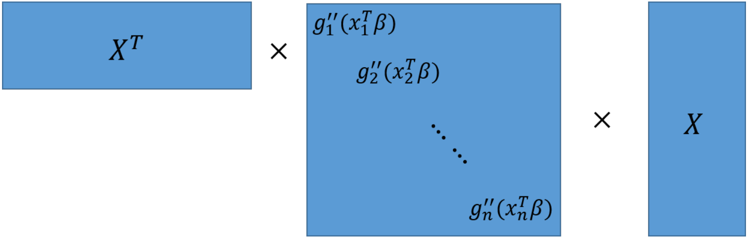
Machine Learning Model

- $f(\beta) = \sum_{i=1}^{n} g_i(x_i^T \beta)$
- $\nabla^2 f(\beta) = X^T \text{diag}\{g_i''(x_i^T \beta)\}_{i=1}^{n} X$



Machine Learning Model

- $f(\beta) = \sum_{i=1}^{n} g_i(x_i^T \beta)$
- $\nabla^2 f(\beta) = X^T \text{diag}\{g_i''(x_i^T \beta)\}_{i=1}^{n} X$



- $\nabla^2 f(\beta)^{1/2} = \text{diag}\{\sqrt{g_i''(x_i^T \beta)}\}_{i=1}^{n} X$

Outline

First Order Method

- Gradient Decent
  $\beta = \beta - \eta \nabla f(\beta)$

First Order Method

- Gradient Decent
  $\beta = \beta - \eta \nabla f(\beta)$
- Stochastic Gradient Descent
  $\beta = \beta - \eta \nabla f_i(\beta)$

First Order Method

- Gradient Decent
  $\beta = \beta - \eta \nabla f(\beta)$
- Stochastic Gradient Descent
  $\beta = \beta - \eta \nabla f_i(\beta)$
- SVRG[Johnson and Zhang, 2013]
  SAG[Schmidt, Roux, and Bach, 2013]
  AGD[Nesterov], . . .

- Gradient Decent
  $\beta = \beta - \eta \nabla f(\beta)$
- Stochastic Gradient Descent
  $\beta = \beta - \eta \nabla f_i(\beta)$
- SVRG[Johnson and Zhang, 2013]
  SAG[Schmidt, Roux, and Bach, 2013]
  AGD[Nesterov], ...
- Remark
  # Iteration(GD) : $\kappa \log(1/\epsilon)$
  (κ : condition number, ϵ : precision)

Outline

Second Order Method

- Newton Method
  $\beta = \beta - \eta[\nabla^2 f(\beta)]^{-1}\nabla f(\beta)$

Second Order Method

- Newton Method
  $\beta = \beta - \eta[\nabla^2 f(\beta)]^{-1}\nabla f(\beta)$
- Quasi Newton Method
  $H_{k+1} = \underset{\text{s.t.} H=H^T, Hy_k=s_k}{\arg\min} \|H - H_k\|$

Second Order Method

- Newton Method
  $\beta = \beta - \eta[\nabla^2 f(\beta)]^{-1}\nabla f(\beta)$
- Quasi Newton Method
  $H_{k+1} = \underset{\text{s.t.} H=H^T, Hy_k=s_k}{\arg\min} \|H - H_k\|$
- Newton-Conjugate Gradient
  $[\nabla^2 f(\beta)]p = -\nabla f(\beta)$

Second Order Method

- Newton Method
  $\beta = \beta - \eta[\nabla^2 f(\beta)]^{-1}\nabla f(\beta)$
- Quasi Newton Method
  $H_{k+1} = \underset{\text{s.t.} H=H^T, Hy_k=s_k}{\arg\min} \|H - H_k\|$
- Newton-Conjugate Gradient
  $[\nabla^2 f(\beta)]p = -\nabla f(\beta)$
- Remark
  $\#$ Iteration(Newton) : $\log\log(1/\epsilon)$
  (ϵ : precision)

Outline

1. Optimization Algorithm Overview
   - Machine Learning Model
   - First Order Method
   - Second Order Method
2. Randomized Second Order Methods
   - Subsampled Newton Method
   - Newton Sketch
3. Our Result
   - Count Sketch + Newton Method
4. Conclusion

Subsampled Newton Method

1. Updates

Subsampled Newton Method

1. Updates
   - Newton method
     $\beta = \beta - \eta[\nabla^2 f(\beta)]^{-1} \nabla f(\beta)$

Subsampled Newton Method

1. Updates
   - Newton method
     $\beta = \beta - \eta[\nabla^2 f(\beta)]^{-1}\nabla f(\beta)$
   - Subsampled Newton[Erdogdu and Montanari, 2015]
     $\nabla^2 f_S(\beta)$ where $S \subset [n]$
     $[\Lambda_{k+1}, U_{k+1}] = \text{T-SVD}(\nabla^2 f_S(\beta), k+1)$[Gavish and Donoho, 2014]
     $\beta = \beta - \eta(\lambda_{k+1}^{-1} I_d + U_k(\Lambda_k^{-1} - \lambda_{k+1}^{-1} I_k)U_k^T)\nabla f(\beta)$

Subsampled Newton Method

1. Updates
   - Newton method
     $\beta = \beta - \eta[\nabla^2 f(\beta)]^{-1}\nabla f(\beta)$
   - Subsampled Newton[Erdogdu and Montanari, 2015]
     $\nabla^2 f_S(\beta)$ where $S \subset [n]$
     $[\Lambda_{k+1}, U_{k+1}] = \text{T-SVD}(\nabla^2 f_S(\beta), k+1)$[Gavish and Donoho, 2014]
     $\beta = \beta - \eta(\lambda_{k+1}^{-1} I_d + U_k(\Lambda_k^{-1} - \lambda_{k+1}^{-1} I_k)U_k^T)\nabla f(\beta)$
2. Complexity

Subsampled Newton Method

1. Updates
   - Newton method
     $\beta = \beta - \eta[\nabla^2 f(\beta)]^{-1}\nabla f(\beta)$
   - Subsampled Newton[Erdogdu and Montanari, 2015]
     $\nabla^2 f_S(\beta)$ where $S \subset [n]$
     $[\Lambda_{k+1}, U_{k+1}] = \text{T-SVD}(\nabla^2 f_S(\beta), k+1)$[Gavish and Donoho, 2014]
     $\beta = \beta - \eta(\lambda_{k+1}^{-1} I_d + U_k(\Lambda_k^{-1} - \lambda_{k+1}^{-1} I_k)U_k^T)\nabla f(\beta)$
2. Complexity
   - Newton method
     $O(nd^2 + d^3)$

Subsampled Newton Method

1. Updates
   - Newton method
     $\beta = \beta - \eta[\nabla^2 f(\beta)]^{-1}\nabla f(\beta)$
   - Subsampled Newton[Erdogdu and Montanari, 2015]
     $\nabla^2 f_S(\beta)$ where $S \subset [n]$
     $[\Lambda_{k+1}, U_{k+1}] = \text{T-SVD}(\nabla^2 f_S(\beta), k+1)$[Gavish and Donoho, 2014]
     $\beta = \beta - \eta(\lambda_{k+1}^{-1} I_d + U_k(\Lambda_k^{-1} - \lambda_{k+1}^{-1} I_k)U_k^T)\nabla f(\beta)$
2. Complexity
   - Newton method
     $O(nd^2 + d^3)$
   - Subsampled Newton
     $O(|S|d^2 + d^2 k + nd)$

Subsampled Newton Method

1. Updates
   - Newton method
     $\beta = \beta - \eta[\nabla^2 f(\beta)]^{-1} \nabla f(\beta)$
   - Subsampled Newton[Erdogdu and Montanari, 2015]
     $\nabla^2 f_S(\beta)$ where $S \subset [n]$
     $[\Lambda_{k+1}, U_{k+1}] = \text{T-SVD}(\nabla^2 f_S(\beta), k+1)$[Gavish and Donoho, 2014]
     $\beta = \beta - \eta(\lambda_{k+1}^{-1} I_d + U_k(\Lambda_k^{-1} - \lambda_{k+1}^{-1} I_k) U_k^T) \nabla f(\beta)$
2. Complexity
   - Newton method
     $O(nd^2 + d^3)$
   - Subsampled Newton
     $O(|S|d^2 + d^2 k + nd)$
   - Accurate approximation to the TSVD
     [Halko, Martinsson, and Tropp, 2011]

Subsampled Newton Method

Convergence rate [Theorem 3.2 [Erdogdu and Montanari, 2015]]

Under the two assumptions of Lipschitz Hessian and Bounded smoothness
(A1)$\|H_S(\beta) - H_S(\beta')\|_2 \leq M_{|S|}\|\beta - \beta'\|_2$
(A2)$\|\nabla^2 f_i(\beta)\|_2 \leq K$
If the step size satisfies $\eta_t \leq \frac{2}{1+\lambda_d^t/\lambda_{r+1}^t}$,
then we have

$$\|\beta^{t+1} - \beta^*\|_2 \leq \xi_1^t\|\beta^t - \beta^*\|_2 + \xi_2^t\|\beta^t - \beta^*\|_2^2$$

w.p. at least $1 - 2/d$
for an absolute constant $c > 0$, for the coefficients ξ_1^t and ξ_2^t are defined as
$\xi_1^t = 1 - \eta_t \frac{\lambda_d^t}{\lambda_{r+1}^t} + \eta_t \frac{cK}{\lambda_{r+1}^t}\sqrt{\frac{\log(d)}{|S_t|}}$ $\xi_2^t = \eta_t \frac{M_n}{2\lambda_{r+1}^t}$

Subsampled Newton Method

Convergence rate

$\|\beta^{t+1} - \beta^*\|_2 \leq \xi_1^t \|\beta^t - \beta^*\|_2 + \xi_2^t \|\beta^t - \beta^*\|_2^2$ w.p. at least $1 - 2/d$

$\xi_1^t = 1 - \eta_t \frac{\lambda_d^t}{\lambda_{r+1}^t} + \eta_t \frac{cK}{\lambda_{r+1}^t}\sqrt{\frac{\log(d)}{|S_t|}}$, $\xi_2^t = \eta_t \frac{M_n}{2\lambda_{r+1}^t}$

1. Convergence rate
   Quadratic-Linear convergence

Subsampled Newton Method

Convergence rate

$\|\beta^{t+1} - \beta^*\|_2 \leq \xi_1^t \|\beta^t - \beta^*\|_2 + \xi_2^t \|\beta^t - \beta^*\|_2^2$ w.p. at least $1 - 2/d$

$\xi_1^t = 1 - \eta_t \frac{\lambda_d^t}{\lambda_{r+1}^t} + \eta_t \frac{cK}{\lambda_{r+1}^t} \sqrt{\frac{\log(d)}{|S_t|}}$, $\xi_2^t = \eta_t \frac{M_n}{2\lambda_{r+1}^t}$

1. Convergence rate
   Quadratic-Linear convergence
2. Stepsize
   step size $\eta_t = 1$ satisfying $\eta_t \leq \frac{2}{1+\lambda_d^t/\lambda_{r+1}^t}$

Subsampled Newton Method

Convergence rate

$\|\beta^{t+1} - \beta^*\|_2 \leq \xi_1^t \|\beta^t - \beta^*\|_2 + \xi_2^t \|\beta^t - \beta^*\|_2^2$ w.p. at least $1 - 2/d$

$\xi_1^t = 1 - \eta_t \frac{\lambda_d^t}{\lambda_{r+1}^t} + \eta_t \frac{cK}{\lambda_{r+1}^t}\sqrt{\frac{\log(d)}{|S_t|}}$, $\xi_2^t = \eta_t \frac{M_n}{2\lambda_{r+1}^t}$

1. Convergence rate
   Quadratic-Linear convergence
2. Stepsize
   step size $\eta_t = 1$ satisfying $\eta_t \leq \frac{2}{1+\lambda_d^t/\lambda_{r+1}^t}$
3. Sub-sample size
   $\xi_1^t = 1 - \eta_t \frac{\lambda_d^t}{\lambda_{r+1}^t} + \eta_t \frac{cK}{\lambda_{r+1}^t}\sqrt{\frac{\log(d)}{|S_t|}}$
   $|S_t| \geq \kappa^2 \log(d)$

Subsampled Newton Method

Convergence rate

$\|\beta^{t+1} - \beta^*\|_2 \leq \xi_1^t \|\beta^t - \beta^*\|_2 + \xi_2^t \|\beta^t - \beta^*\|_2^2$ w.p. at least $1 - 2/d$

$\xi_1^t = 1 - \eta_t \frac{\lambda_d^t}{\lambda_{r+1}^t} + \eta_t \frac{cK}{\lambda_{r+1}^t} \sqrt{\frac{\log(d)}{|S_t|}}$, $\xi_2^t = \eta_t \frac{M_n}{2\lambda_{r+1}^t}$

1. Convergence rate
   Quadratic-Linear convergence
2. Stepsize
   step size $\eta_t = 1$ satisfying $\eta_t \leq \frac{2}{1+\lambda_d^t/\lambda_{r+1}^t}$
3. Sub-sample size
   $\xi_1^t = 1 - \eta_t \frac{\lambda_d^t}{\lambda_{r+1}^t} + \eta_t \frac{cK}{\lambda_{r+1}^t} \sqrt{\frac{\log(d)}{|S_t|}}$
   $|S_t| \geq \kappa^2 \log(d)$
4. Extension
   Non-uniform sampling(sampling through the leverage score)
   [Xu, Yang, Roosta-Khorasani, Ré, and Mahoney, 2016]

Outline

Newton Sketch

- Newton Sketch
  $\beta = \beta - \eta[\nabla^2 f(\beta)^{1/2T} S^T S \nabla^2 f(\beta)^{1/2}]^{-1} \nabla f(\beta)$

Newton Sketch

- Newton Sketch
  $\beta = \beta - \eta[\nabla^2 f(\beta)^{1/2T} S^T S \nabla^2 f(\beta)^{1/2}]^{-1} \nabla f(\beta)$
- Sub-sampled Newton
  $\beta = \beta - \eta(\lambda_{k+1}^{-1} I_d + U_k(\Lambda_k^{-1} - \lambda_{k+1}^{-1} I_k) U_k^T) \nabla f(\beta)$

Newton Sketch

- Newton Sketch
  $\beta = \beta - \eta[\nabla^2 f(\beta)^{1/2\,T} S^T S \nabla^2 f(\beta)^{1/2}]^{-1} \nabla f(\beta)$
- Sub-sampled Newton
  $\beta = \beta - \eta(\lambda_{k+1}^{-1} I_d + U_k(\Lambda_k^{-1} - \lambda_{k+1}^{-1} I_k) U_k^T) \nabla f(\beta)$
- Newton Method
  $\beta = \beta - \eta[\nabla^2 f(\beta)]^{-1} \nabla f(\beta)$

Newton Sketch

- Update : $\beta = \beta - \eta[\nabla^2 f(\beta)^{1/2\,T} S^T S \nabla^2 f(\beta)^{1/2}]^{-1} \nabla f(\beta)$

Newton Sketch

- Update : $\beta = \beta - \eta[\nabla^2 f(\beta)^{1/2\,T} S^T S \nabla^2 f(\beta)^{1/2}]^{-1} \nabla f(\beta)$
- Sketch matrix $S \in \mathcal{R}^{m \times n}$

Newton Sketch

- Update : $\beta = \beta - \eta[\nabla^2 f(\beta)^{1/2\,T} S^T S \nabla^2 f(\beta)^{1/2}]^{-1} \nabla f(\beta)$
- Sketch matrix $S \in \mathcal{R}^{m \times n}$
  1. Random Projection
     S has i.i.d. standard Gaussian entries
     Time cost: $O(nmd)$ and $m = O(d/\epsilon^2)$

Newton Sketch

- Update : $\beta = \beta - \eta[\nabla^2 f(\beta)^{1/2T} S^T S \nabla^2 f(\beta)^{1/2}]^{-1} \nabla f(\beta)$
- Sketch matrix $S \in \mathcal{R}^{m \times n}$
  1. Random Projection
     S has i.i.d. standard Gaussian entries
     Time cost: $O(nmd)$ and $m = O(d/\epsilon^2)$
  2. Subsampled Random Hadamard
     $S = PHD$
     D : diagonal matrix with entries sampled uniformly from $\{+1, -1\}$
     H : Hadamard and P : samples m from n rows
     Time cost: $O(nd \log m)$ and $m = O((d + \log n) \log d/\epsilon^2)$[Woodruff et al., 2014]

Newton Sketch

- Update : $\beta = \beta - \eta[\nabla^2 f(\beta)^{1/2\,T} S^T S \nabla^2 f(\beta)^{1/2}]^{-1} \nabla f(\beta)$
- Sketch matrix $S \in \mathcal{R}^{m \times n}$
  1. Random Projection
     S has i.i.d. standard Gaussian entries
     Time cost: $O(nmd)$ and $m = O(d/\epsilon^2)$
  2. Subsampled Random Hadamard
     $S = PHD$
     D : diagonal matrix with entries sampled uniformly from $\{+1, -1\}$
     H : Hadamard and P : samples m from n rows
     Time cost: $O(nd \log m)$ and $m = O((d + \log n) \log d/\epsilon^2)$[Woodruff et al., 2014]
  3. Random Row Sampling
     S : samples m from n rows
     same with subsampled Newton method
     $m = O(\kappa^2 \log d)$

Newton Sketch

Local Convergence[Theorem 1 [Pilanci and Wainwright, 2015]]

For given parameters $\delta, \epsilon \in (0,1)$, consider the Newton sketch updates based on an initialization $\beta^{(0)}$ such that $\|\beta^{(0)} - \beta^*\|_2 \leq \delta \frac{a}{8L}$ and a sketch dimension m satisfying the lower bound $\frac{c}{\epsilon^2} n$. Then with probabilty at least $1 - c_1 \exp(-c_2 m)$, the l_2-error satisfies the recusrsion

$$\|\beta^{(t+1)} - \beta^*\|_2 \leq \epsilon \frac{b}{a} \|\beta^{(t)} - \beta^*\|_2 + \frac{4L}{a} \|\beta^{(t)} - \beta^*\|_2^2$$

a and b are strongly convexity and smoothness of the function f at the optimal point β^*, and L is hessian Lipshitz

Outline

Our Result

1. Random Projection
   $O(nmd)$
   $m = O(d/\epsilon^2)$

Our Result

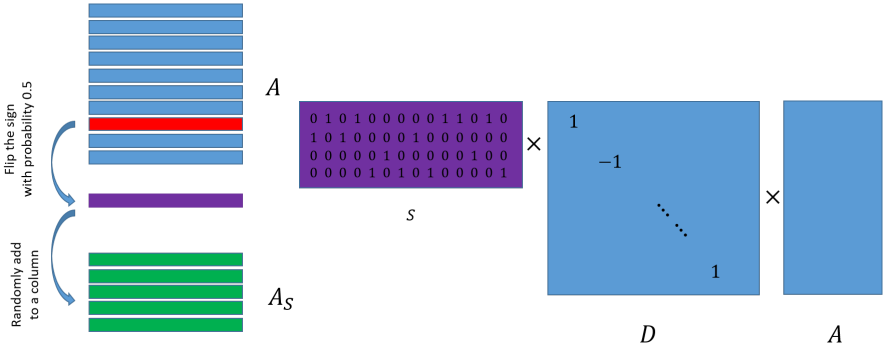
1. Random Projection
   $O(nmd)$
   $m = O(d/\epsilon^2)$
2. Subsample Random Hadamard
   $O(nd \log m)$
   $m = O((d + \log n) \log d/\epsilon^2)$

Our Result

1. Random Projection
   $O(nmd)$
   $m = O(d/\epsilon^2)$
2. Subsample Random Hadamard
   $O(nd \log m)$
   $m = O((d + \log n) \log d/\epsilon^2)$
3. CountSketch[Clarkson and Woodruff, 2013]
   $O(\mathrm{nnz}(X))$
   $m = O(d^2/\epsilon^2)$

Our Result

- CountSketch: $O(\text{nnz}(A))$



Our Result

1. Combined
   CountSketch(m_{CS})+Random Projection(m): $O(\mathrm{nnz}(X) + m_{\mathrm{CS}} dm)$
   CountSketch(m_{CS})+SRHT(m): $O(\mathrm{nnz}(X) + m_{\mathrm{CS}} d \log m)$

Our Result

1. Combined
   CountSketch(m_{CS})+Random Projection(m): $O(\text{nnz}(X) + m_{\text{CS}} dm)$
   CountSketch(m_{CS})+SRHT(m): $O(\text{nnz}(X) + m_{\text{CS}} d \log m)$
2. To do
   CountSketch + Newton Method $\Rightarrow$ Quadratic-Linear Convergence
   (1) Test empirically
   (2) Prove Q-L convergence rate with CountSketch

Outline

Conclusion

- There was a revival of second order methods by using a sketching and a low-rank approximation in the last year

Conclusion

- There was a revival of second order methods by using a sketching and a low-rank approximation in the last year
- Introduce other three papers regarding to second order method.

Conclusion

- There was a revival of second order methods by using a sketching and a low-rank approximation in the last year
- Introduce other three papers regarding to second order method.
  1. [Luo, Agarwal, Cesa-Bianchi, and Langford, 2016] deal with second order online learning enhancing online Newton step

Conclusion

- There was a revival of second order methods by using a sketching and a low-rank approximation in the last year
- Introduce other three papers regarding to second order method.
  1. [Luo, Agarwal, Cesa-Bianchi, and Langford, 2016] deal with second order online learning enhancing online Newton step
  2. [Agarwal, Bullins, and Hazan, 2016] use the property of $A^{-1} = \sum_{i=0}^{\infty}(I - A)^i$ to get hessian inversion efficiently

Conclusion

- There was a revival of second order methods by using a sketching and a low-rank approximation in the last year
- Introduce other three papers regarding to second order method.
    1. [Luo, Agarwal, Cesa-Bianchi, and Langford, 2016] deal with second order online learning enhancing online Newton step
    2. [Agarwal, Bullins, and Hazan, 2016] use the property of $A^{-1} = \sum_{i=0}^{\infty}(I - A)^i$ to get hessian inversion efficiently
    3. [Gower, Goldfarb, and Richtárik, 2016] apply sketching ideas to BFGS algorithm

Thank you

References I

Naman Agarwal, Brian Bullins, and Elad Hazan. Second order stochastic optimization in linear time. *arXiv preprint arXiv:1602.03943*, 2016.

Kenneth L Clarkson and David P Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 81–90. ACM, 2013.

Murat A Erdogdu and Andrea Montanari. Convergence rates of sub-sampled newton methods. In *Advances in Neural Information Processing Systems*, pages 3052–3060, 2015.

Matan Gavish and David L Donoho. Optimal shrinkage of singular values. *arXiv preprint arXiv:1405.7511*, 2014.

Robert M Gower, Donald Goldfarb, and Peter Richtárik. Stochastic block bfgs: Squeezing more curvature out of data. *arXiv preprint arXiv:1603.09649*, 2016.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

References II

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

Haipeng Luo, Alekh Agarwal, Nicolò Cesa-Bianchi, and John Langford. Efficient second order online learning by sketching. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 902–910. Curran Associates, Inc., 2016. URL `http://papers.nips.cc/paper/6207-efficient-second-order-online-learning-by-sketching.pdf`.

Yurii Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2).

Mert Pilanci and Martin J Wainwright. Newton sketch: A linear-time optimization algorithm with linear-quadratic convergence. *arXiv preprint arXiv:1505.02250*, 2015.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *arXiv preprint arXiv:1309.2388*, 2013.

David P Woodruff et al. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1–2):1–157, 2014.

Peng Xu, Jiyan Yang, Farbod Roosta-Khorasani, Christopher Ré, and Michael W Mahoney. Sub-sampled newton methods with non-uniform sampling. *arXiv preprint arXiv:1607.00559*, 2016.