Overcoming Inversion Bias in Distributed Newton's Method

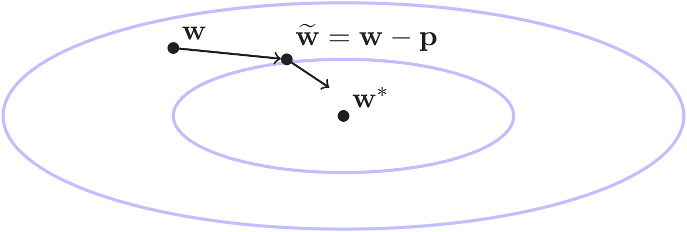
Michael W. Mahoney

ICSI and Department of Statistics
University of California, Berkeley

Joint work with Burak Bartan, Mert Pilanci, and **Michał Dereziński**

Convex optimization

$$\text{Find} \quad \mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \mathcal{L}(\mathbf{w})$$



Why use second-order methods...

...when there is SGD?

- Sensitive to hyper-parameters
- Limited effectiveness for large batch training

See, e.g., [DMK$^+$18, GVY$^+$18]

Second-order:

- No hyper-parameter tuning
- Supports large batch training

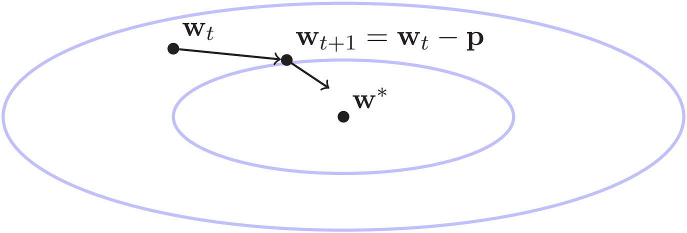
Recent interest in second-order methods:

- theoretical analysis [RKM19, RLXM18, WRKXM18]
- empirical (including DNNs) [GKC$^+$19, FKR$^+$18, KRMG18]

Newton's method

$$\mathcal{L}(\mathbf{w}) \;=\; \frac{1}{n}\sum_{j=1}^{n} \ell_j(\mathbf{w}^\top \mathbf{x}_j) \;+\; \frac{\lambda}{2}\|\mathbf{w}\|^2$$

$$\mathbf{p} = \Big[\; \underbrace{\nabla^2\mathcal{L}(\mathbf{w})}_{\text{Hessian } \mathbf{H}} \;\Big]^{-1} \underbrace{\nabla\mathcal{L}(\mathbf{w})}_{\text{gradient } \mathbf{g}}$$

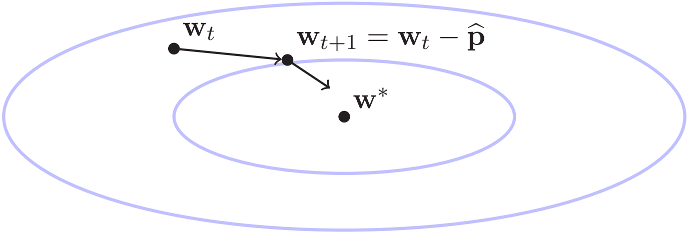


Newton's method

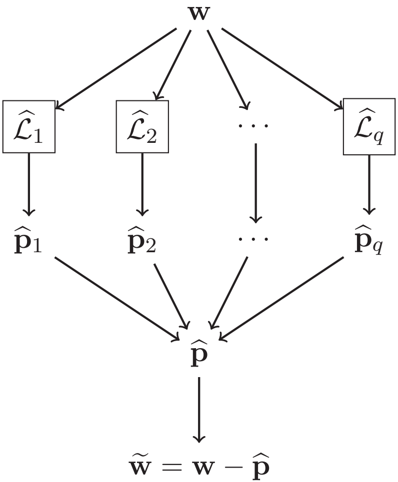
Approximate Newton's method

$$\widehat{\mathcal{L}}(\mathbf{w}) \;=\; \frac{1}{m}\sum_{j\in S}\ell_j(\mathbf{w}^\top \mathbf{x}_j) \;+\frac{\lambda}{2}\|\mathbf{w}\|^2$$

$$\widehat{\mathbf{p}} = \Big[\; \underbrace{\nabla^2\widehat{\mathcal{L}}(\mathbf{w})}_{\text{Hessian estimate } \widehat{\mathbf{H}}} \;\Big]^{-1} \underbrace{\nabla\mathcal{L}(\mathbf{w})}_{\text{gradient } \mathbf{g}}$$



Distributed Newton's method



Question: How to combine local Newton estimates $\widehat{\mathbf{p}}_1, ..., \widehat{\mathbf{p}}_q$?

Model averaging

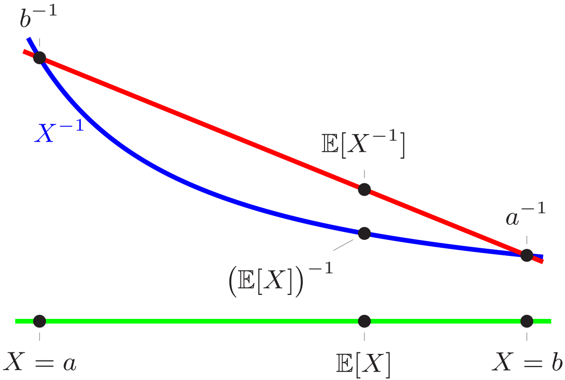
Standard averaging leads to biased estimates:

$$\lim_{q\to\infty} \frac{1}{q}\sum_{t=1}^{q} \widehat{\mathbf{p}}_t \;\neq\; \mathbf{p} \qquad (q \text{ is the number of machines})$$

$$\mathbb{E}\big[\widehat{\mathbf{H}}^{-1}\big] \neq \mathbf{H}^{-1}, \qquad \text{even though} \quad \mathbb{E}\big[\widehat{\mathbf{H}}\big] = \mathbf{H}.$$

General phenomenon: Inversion bias

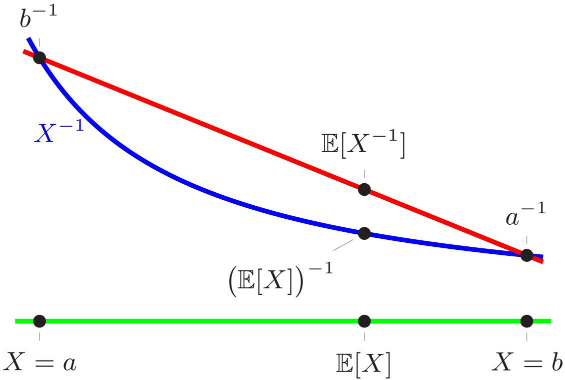
Inversion bias: $\mathbb{E}[X^{-1}] \neq \big(\mathbb{E}[X]\big)^{-1}$ for random X



General phenomenon: Inversion bias

$$\text{Inversion bias:} \quad \mathbb{E}[X^{-1}] \neq \big(\mathbb{E}[X]\big)^{-1} \quad \text{for random } X$$

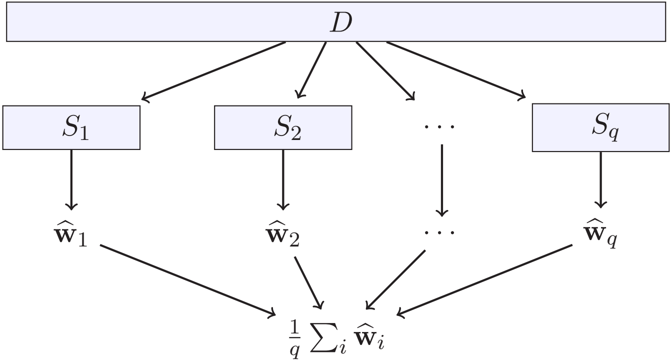
Extends to inverting high-dimensional random matrices



Inversion bias in model averaging

1. Bagging
2. Distributed optimization
3. Federated learning

$$\mathbf{w}^* - \frac{1}{q}\sum_i \widehat{\mathbf{w}}_i \xrightarrow{q\to\infty} \underbrace{\mathbf{w}^* - \mathbb{E}[\widehat{\mathbf{w}}_i]}_{\text{Bias}}$$



Determinantal correction

Hessian estimate: $\widehat{\mathbf{H}} = \nabla^2 \widehat{\mathcal{L}}(\mathbf{w})$

Inversion bias: $\mathbb{E}\big[\widehat{\mathbf{H}}^{-1}\big] \neq \mathbf{H}^{-1}$

Determinantal correction

Hessian estimate: $\widehat{\mathbf{H}} = \nabla^2 \widehat{\mathcal{L}}(\mathbf{w})$

Inversion bias: $\mathbb{E}\big[\widehat{\mathbf{H}}^{-1}\big] \neq \mathbf{H}^{-1}$

Correction: $\dfrac{\mathbb{E}\big[\det(\widehat{\mathbf{H}})\widehat{\mathbf{H}}^{-1}\big]}{\mathbb{E}\big[\det(\widehat{\mathbf{H}})\big]} = \mathbf{H}^{-1}$

Determinantal correction

Hessian estimate: $\widehat{\mathbf{H}} = \nabla^2 \widehat{\mathcal{L}}(\mathbf{w})$

Inversion bias: $\mathbb{E}\big[\widehat{\mathbf{H}}^{-1}\big] \neq \mathbf{H}^{-1}$

Correction: $$\frac{\mathbb{E}\big[\det(\widehat{\mathbf{H}})\widehat{\mathbf{H}}^{-1}\big]}{\mathbb{E}\big[\det(\widehat{\mathbf{H}})\big]} = \mathbf{H}^{-1}$$

Two strategies of using the correction:

1. Weighted averaging instead of uniform averaging
   Determinantal averaging [**D**M19]
2. Joint sampling instead of uniform sampling
   Surrogate sketches [**D**BPM20]

Comparison of two strategies

Determinantal averaging

- consistent global estimate: $\widehat{\mathbf{p}} \underset{m\to\infty}{\longrightarrow} \mathbf{p}$
- works with uniform sampling

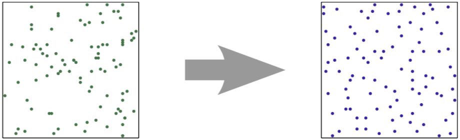
Surrogate sketching

- unbiased local estimates: $\mathbb{E}[\widehat{\mathbf{p}}_t] = \mathbf{p}$
- samples from a Determinantal Point Process (DPP)

Determinantal Point Processes (DPPs)

Non-i.i.d. randomized selection of a data subset S

Negative correlation: $\Pr(i \in S \mid j \in S) < \Pr(i \in S)$



i.i.d. (left) versus DPP (right)

- Fast algorithms: [C**D**V20] (NeurIPS'20)

  "Sampling from a k-DPP without looking at all items"

- Learn more: [**D**M20] (Notices of the AMS)

  "Determinantal point processes in randomized numerical linear algebra"

Image from [KT12]

Correcting inversion bias in Distributed Newton

Baseline: Uniform averaging of biased estimates [WRKXM18]

Convergence rate: $\|\mathbf{w}_{t+1} - \mathbf{w}^*\| = \widetilde{O}\Big(\sqrt{\frac{d}{qm}} + \frac{d}{m} \Big) \cdot \|\mathbf{w}_t - \mathbf{w}^*\|$

"variance" *"bias"*

q - number of machines
m - data points per machine
d - number of parameters

Method	Convergence rate	Trade-offs
Baseline	$\sqrt{\frac{d}{qm}} + \frac{d}{m}$	Var Bias Cost

[**D**M19] *"Distributed estimation of the inverse Hessian by determinantal averaging"*, at NeurIPS'19.

Correcting inversion bias in Distributed Newton

Baseline: Uniform averaging of biased estimates [WRKXM18]

Convergence rate: $\|\mathbf{w}_{t+1} - \mathbf{w}^*\| = \widetilde{O}\Big(\underbrace{\sqrt{\tfrac{d}{qm}}}_{\text{“variance”}} + \underbrace{\tfrac{d}{m}}_{\text{“bias”}} \Big) \cdot \|\mathbf{w}_t - \mathbf{w}^*\|$

q - number of machines
m - data points per machine
d - number of parameters

	Method	Convergence rate	Trade-offs
	Baseline	$\sqrt{\frac{d}{qm}} + \frac{d}{m}$	Var, **Bias**, Cost
[**D**M19]	*Determinantal averaging*	$\frac{d}{\sqrt{qm}}$	**Var**, Bias, Cost

[**D**M19] *“Distributed estimation of the inverse Hessian by determinantal averaging”*, at NeurIPS’19.

Correcting inversion bias in Distributed Newton

Baseline: Uniform averaging of biased estimates [WRKXM18]

Convergence rate: $\|\mathbf{w}_{t+1} - \mathbf{w}^*\| = \widetilde{O}\Big(\underbrace{\sqrt{\tfrac{d}{qm}}}_{\text{"variance"}} + \underbrace{\tfrac{d}{m}}_{\text{"bias"}} \Big) \cdot \|\mathbf{w}_t - \mathbf{w}^*\|$

q - number of machines
m - data points per machine
d - number of parameters

	Method	Convergence rate	Trade-offs
	Baseline	$\sqrt{\frac{d}{qm}} + \frac{d}{m}$	Var, **Bias**, Cost
[**D**M19]	*Determinantal averaging*	$\frac{d}{\sqrt{qm}}$	**Var**, Bias, Cost
[**D**BPM20]	*Surrogate sketching*	$\sqrt{\frac{d}{qm}}$	Var, Bias, **Cost**

[**D**BPM20] *"Debiasing distributed second order optimization with surrogate sketching and scaled regularization"*, at NeurIPS'20.

Correcting inversion bias in Distributed Newton

Baseline: Uniform averaging of biased estimates [WRKXM18]

Convergence rate: $\|\mathbf{w}_{t+1} - \mathbf{w}^*\| = \widetilde{O}\Big(\underbrace{\sqrt{\tfrac{d}{qm}}}_{\text{"variance"}} + \underbrace{\tfrac{d}{m}}_{\text{"bias"}} \Big) \cdot \|\mathbf{w}_t - \mathbf{w}^*\|$

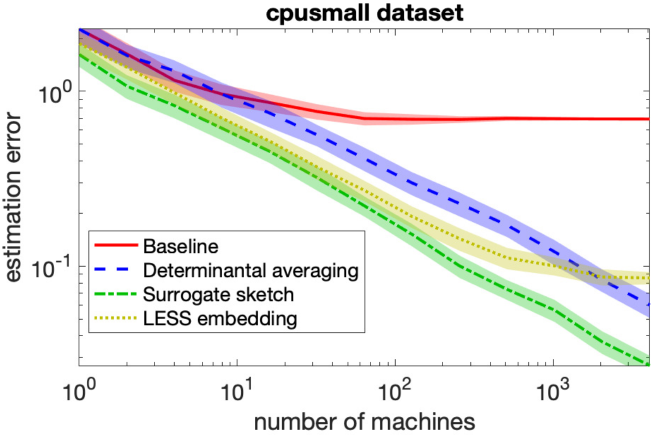
q - number of machines
m - data points per machine
d - number of parameters

	Method	Convergence rate	Trade-offs
	Baseline	$\sqrt{\frac{d}{qm}} + \frac{d}{m}$	Var Bias Cost
[DM19]	*Determinantal averaging*	$\frac{d}{\sqrt{qm}}$	Var Bias Cost
[DBPM20]	*Surrogate sketching*	$\sqrt{\frac{d}{qm}}$	Var Bias Cost
[DLDM20]	*LESS embeddings*	$\sqrt{\frac{d}{qm}} + \frac{\sqrt{d}}{m}$	Var Bias Cost

[DLDM20] *"Sparse sketches with small inversion bias"*, Preprint at arXiv:2011.10695.

Bias-variance trade-offs in model averaging

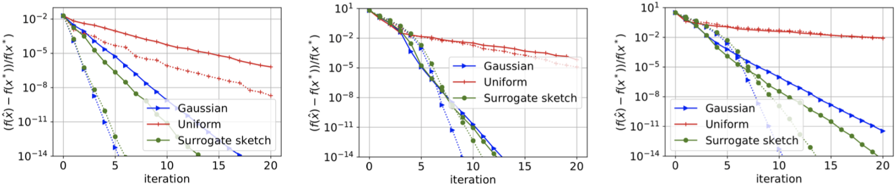
$$\text{estimation error} \quad = \quad \Big\| \frac{1}{q}\sum_{i=1}^{q} \widehat{\mathbf{p}}_i - \mathbf{p}^* \Big\|$$



Experiments: Effect of implicit regularization

Question: Should local regularizer match the global λ?

Distributed Newton with 100 machines for logistic regression



(a) statlog-australian-credit (b) breast-cancer-wisc (c) ionosphere

dashed lines: *local regularizer* $= \lambda \cdot \left(1 - \frac{d_\lambda}{m}\right)$

$$\text{regularized loss:} \quad \mathcal{L}(\mathbf{w}) = \frac{1}{n}\sum_{j=1}^{n} \ell_j(\mathbf{w}^\top \mathbf{x}_j) + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

[**D**BPM20] *"Debiasing distributed second order optimization with surrogate sketching and scaled regularization"*, at NeurIPS'20.

Conclusions

- Distributed Newton's method suffers from <u>inversion bias</u>
- We can correct this bias with:
  - <u>Weighted averaging</u> instead of uniform averaging
    Determinantal averaging
  - <u>Joint sampling</u> instead of uniform sampling
    Surrogate sketches
  - <u>Scaled local regularization</u> in place of the global regularizer
    $\lambda' = \lambda \cdot (1 - \frac{d_\lambda}{m})$

Thank you!

Daniele Calandriello, Michał Dereziński, and Michal Valko.
Sampling from a k-dpp without looking at all items.
In Conference on Neural Information Processing Systems, 2020.

Michał Dereziński, Burak Bartan, Mert Pilanci, and Michael W Mahoney.
Debiasing distributed second order optimization with surrogate sketching and scaled regularization.
In Conference on Neural Information Processing Systems, 2020.

Michał Dereziński, Zhenyu Liao, Edgar Dobriban, and Michael W Mahoney.
Sparse sketches with small inversion bias.
arXiv preprint arXiv:2011.10695, 2020.

Michał Dereziński and Michael W Mahoney.
Distributed estimation of the inverse hessian by determinantal averaging.
In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems 32, pages 11401–11411. Curran Associates, Inc., 2019.

Michał Dereziński and Michael W Mahoney.
Determinantal point processes in randomized numerical linear algebra.
Notices of the American Mathematical Society, 68(1):34–45, 2021.

References II

Michał Dereziński, Dhruv Mahajan, S. Sathiya Keerthi, S. V. N. Vishwanathan, and Markus Weimer.
Batch-expansion training: An efficient optimization framework.
In Amos Storkey and Fernando Perez-Cruz, editors, Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, volume 84 of Proceedings of Machine Learning Research, pages 736–744, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018.

Chih-Hao Fang, Sudhir B Kylasa, Fred Roosta, Michael W. Mahoney, and Ananth Grama.
Newton-ADMM: A Distributed GPU-Accelerated Optimizer for Multiclass Classification Problems.
arXiv e-prints, page arXiv:1807.07132, Jul 2018.

Vipul Gupta, Swanand Kadhe, Thomas Courtade, Michael W. Mahoney, and Kannan Ramchandran.
OverSketched Newton: Fast Convex Optimization for Serverless Systems.
arXiv e-prints, page arXiv:1903.08857, Mar 2019.

Noah Golmant, Nikita Vemuri, Zhewei Yao, Vladimir Feinberg, Amir Gholami, Kai Rothauge, Michael W. Mahoney, and Joseph Gonzalez.
On the Computational Inefficiency of Large Batch Sizes for Stochastic Gradient Descent.
arXiv e-prints, page arXiv:1811.12941, Nov 2018.

- Sudhir B. Kylasa, Farbod Roosta-Khorasani, Michael W. Mahoney, and Ananth Grama.
GPU Accelerated Sub-Sampled Newton's Method.
arXiv e-prints, page arXiv:1802.09113, Feb 2018.
- Alex Kulesza and Ben Taskar.
Determinantal Point Processes for Machine Learning.
Now Publishers Inc., Hanover, MA, USA, 2012.
- Farbod Roosta-Khorasani and Michael W. Mahoney.
Sub-sampled newton methods.
Math. Program., 174(1–2):293–326, March 2019.
- Fred Roosta, Yang Liu, Peng Xu, and Michael W. Mahoney.
Newton-MR: Newton's Method Without Smoothness or Convexity.
arXiv e-prints, page arXiv:1810.00303, Sep 2018.
- Shusen Wang, Farbod Roosta-Khorasani, Peng Xu, and Michael W Mahoney.
GIANT: Globally improved approximate newton method for distributed optimization.
In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, Advances in Neural Information Processing Systems 31, pages 2332–2342. Curran Associates, Inc., 2018.