We should all run HOGWILD!

Benjamin Recht
Department of EECS and Stats
University of California, Berkeley

with Feng Niu
Christopher Ré
Stephen Wright

Is SGD inherently Serial?

- How to parallelize SGD?
  - Master/Worker (Bertsekas and Tsitsiklis 1985)
  - Round Robin (Langford et al, 2009)
  - Average Runs (Zinkevich et al, 2010)
  - Average Gradients (Duchi et al, Xiao et al 2010)

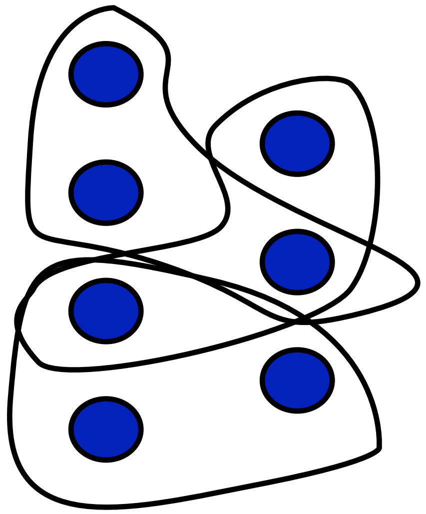
- All require massive overhead due to lock contention and synchronization

- Don’t lock! Don’t communicate!

What happens when we run parallel instances of SGD without locks?

“Sparse” Function: $f(x) = \sum_{e \in E} f_e(x_e)$

- Hypergraph: *G* = (*V*,*E*)
  - *V* - coordinates on $\mathbb{R}^D$
  - *E* - *v* is in *e*∈*E* if f_e depends on x_v



- Graph statistics:

$$\Omega := \max_{e \in E} |e|$$

Maximum Edge Size

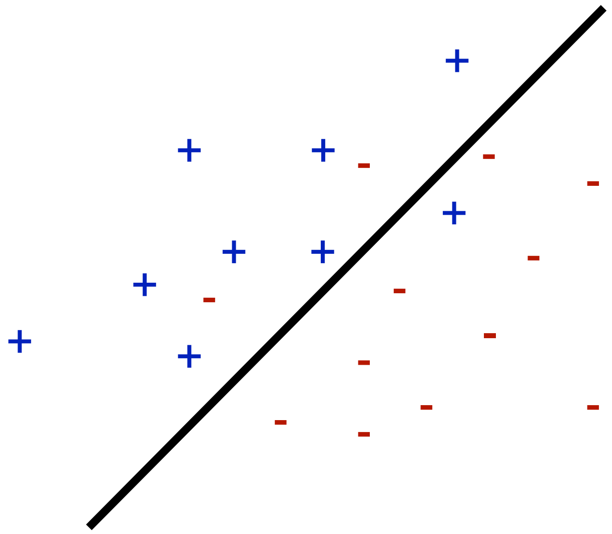
$$\Delta := \frac{\max_{1 \le v \le n} |\{e \in E \ : \ v \in e\}|}{|E|}$$

Maximum Degree

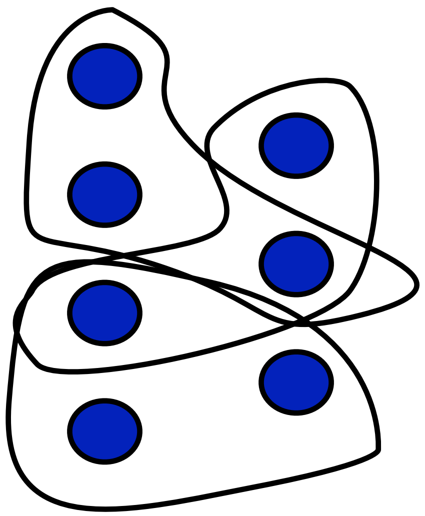
$$\rho := \frac{\max_{e \in E} |\{\hat{e} \in E \ : \hat{e} \cap e \neq \emptyset\}|}{|E|}$$

Maximum Edge Degree

Sparse Support Vector Machines



sparse vectors



$$\text{minimize}_x \sum_{\alpha \in E} \max(1 - y_\alpha x^T z_\alpha, 0) + \lambda \|x\|_2^2$$

$$\text{minimize}_x \sum_{\alpha \in E} \left(\max(1 - y_\alpha x^T z_\alpha, 0) + \lambda \sum_{u \in e_\alpha} \frac{x_u^2}{d_u} \right)$$

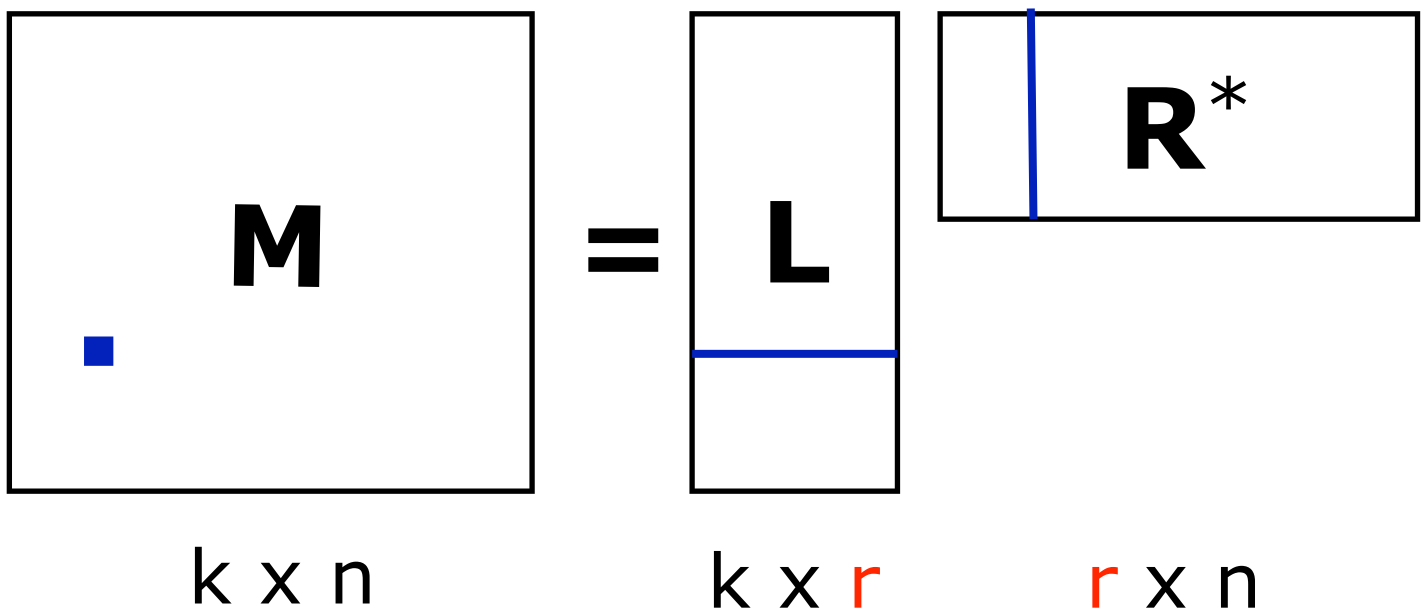
edge degrees

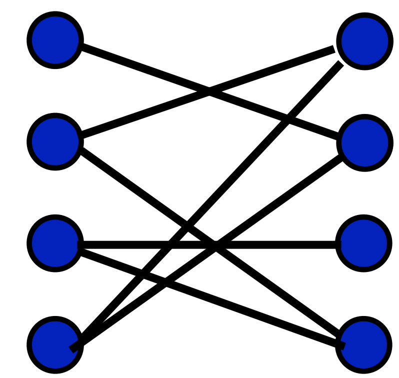
$\Omega = \max_\alpha \|z_\alpha\|_0$

$\Delta = \max_u d_u / D$

$\rho \in (0, 1]$

Matrix Completion





Entries Specified on set *E*

$$\text{minimize} \quad \sum_{(u,v)\in E} (X_{uv} - M_{uv})^2 + \mu \|\mathbf{X}\|_*$$

Idea: approximate $\mathbf{X} \approx \mathbf{L}\mathbf{R}^T$

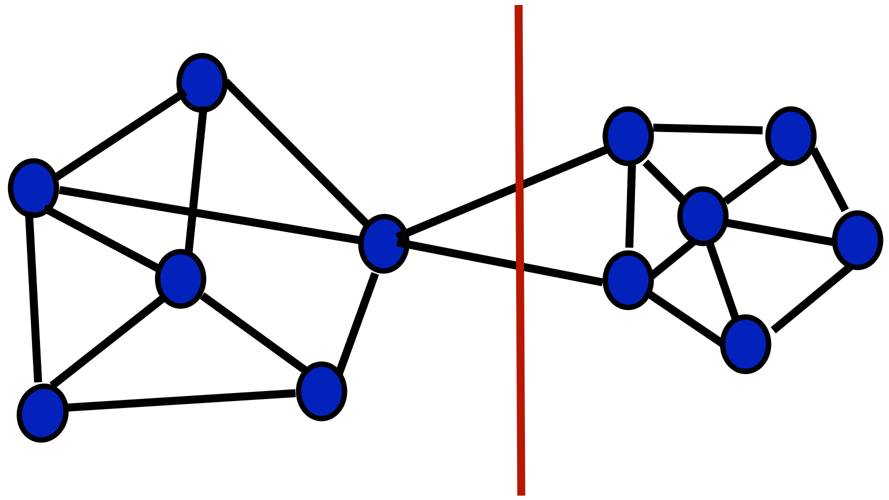
$$\text{minimize}_{(\mathbf{L},\mathbf{R})} \sum_{(u,v)\in E} \left\{ (\mathbf{L}_u \mathbf{R}_v^T - M_{uv})^2 + \mu_u \|\mathbf{L}_u\|_F^2 + \mu_v \|\mathbf{R}_v\|_F^2 \right\}$$

$\Omega = 2r$

$\Delta = O(\log(n)/n)$

$\rho = O(\log(n)/n)$

Graph Cuts



- Image Segmentation
- Entity Resolution
- Topic Modeling

$$\begin{array}{ll} \text{minimize}_x & \sum_{(u,v)\in E} w_{uv} \|x_u - x_v\|_1 \\ \text{subject to} & \mathbf{1}_K^T x_v = 1\,,\ x_v \geq 0\,, \quad \text{for } v = 1, \ldots, D \end{array}$$

$$\Omega = 2K$$

$$\Delta = d_{\max}/|E|$$

$$\rho = 2d_{\max}/|E|$$

HOGWILD!

Run SGD in parallel without locks.

Each processor independently runs:

1. Sample e from E
2. Read current state of x_e
3. **for** v in e **do** $x_v \leftarrow x_v - \alpha[\nabla f_e(x_e)]_v$

Only assume atomicity of $x_v \leftarrow x_v - a$

Issues:

- Updates can be very old
- Processors can overwrite each others' work

Convergence Theory

$$\Omega := \max_{e \in E} |e|$$

$$\Delta := \frac{\max_{1 \leq v \leq n} |\{e \in E \, : \, v \in e\}|}{|E|}$$

$$\rho := \frac{\max_{e \in E} |\{\hat{e} \in E \ : \hat{e} \cap e \neq \emptyset\}|}{|E|}$$

ASSUME:

$$cI \preceq \nabla^2 f \preceq LI$$

$$||\nabla f_e(x)||_2 \leq M$$

$$D_0 = \|x_0 - x_{\text{opt}}\|_2$$

ASSUME: Longest delay between an update and a memory read is τ

Choose: $$k \geq \frac{2LM^2 \left(1 + 6\tau\rho + 6\tau^2 \Omega \Delta^{1/2}\right) \log(LD_0/\epsilon)}{c^2 \epsilon}$$

Then after *k* gradient updates with appropriate choice of *constant* stepsize, we have

$$\mathbb{E}[\|x_k - x_{\text{opt}}\|_2] \leq \epsilon$$

Robust 1/k rates

Nemirovski *et al* (2009): $\gamma_k = \frac{\Theta}{2ck}$ with $\Theta > 1$

$$\|x_k - x_{\mathrm{opt}}\| \leq \frac{1}{k} \max\left\{ \frac{M^2}{c^2} \cdot \frac{\Theta^2}{4\Theta - 4}, D_0 \right\}$$

- If Θ<1, can get exponentially slow convergence
- Slow rate if D0 is large

Sort of obvious, but…

$$\gamma_k = \frac{\vartheta}{2c} \text{ with } \vartheta < 1\,, \quad \text{reduce by } \beta \text{ after } \frac{\log(2/\beta)}{\vartheta\beta^k} \text{ iterations}$$

$$\|x_k - x_{\mathrm{opt}}\| \leq \frac{\log(2/\beta)}{4(1-\beta)} \cdot \frac{M^2}{c^2} \cdot \frac{1}{k - \vartheta^{-1} \log\left(\frac{4D_0c^2}{\vartheta M^2}\right)}$$

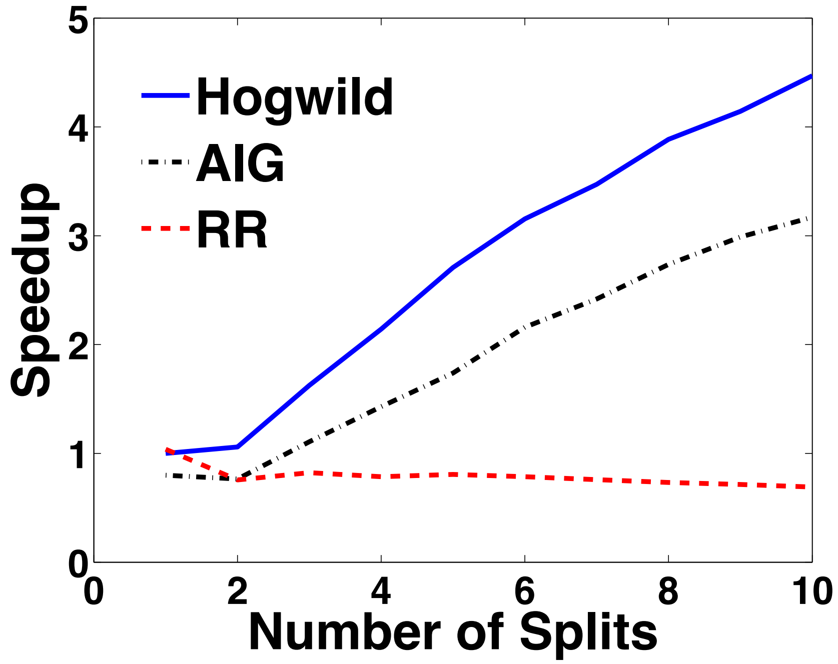
- Pay linearly for bad curvature estimate
- Logarithmic dependence on D0

Hogs gone wild!

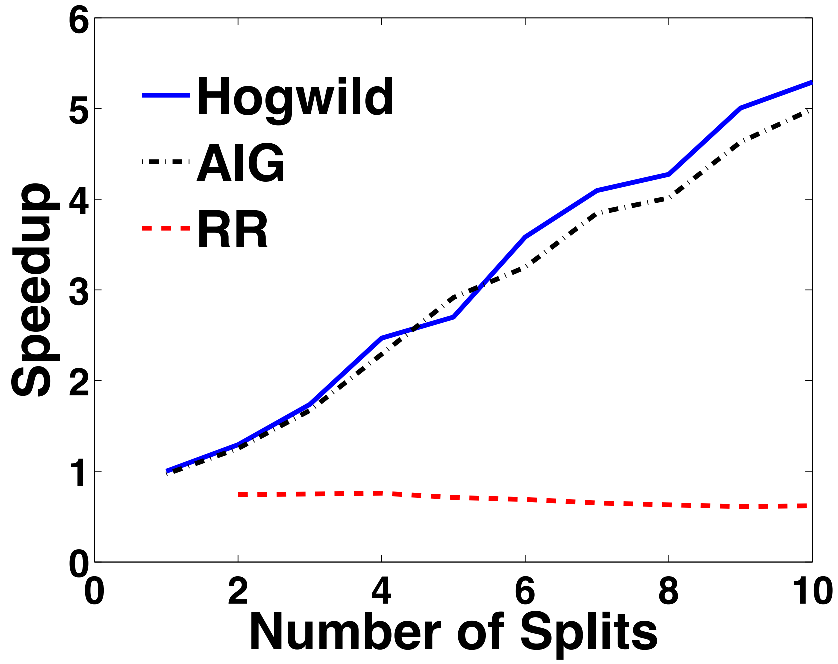
	data set	size (GB)	ρ	Δ	time (s)	speedup
SVM	RCV1	0.9	4.4E-01	1.0E+00	10	4.5
MC	Netflix	1.5	2.5E-03	2.3E-03	301	5.3
	KDD	3.9	3.0E-03	1.8E-03	878	5.2
	JUMBO	30	2.6E-07	1.4E-07	9,454	6.8
CUTS	DBLife	0.003	8.6E-03	4.3E-03	230	8.8
	Abdomen	18	9.2E-04	9.2E-04	1,181	4.1

Experiments run on 12 core machine
All times are for 20 epochs
10 cores for gradients, 2 cores for data shuffling

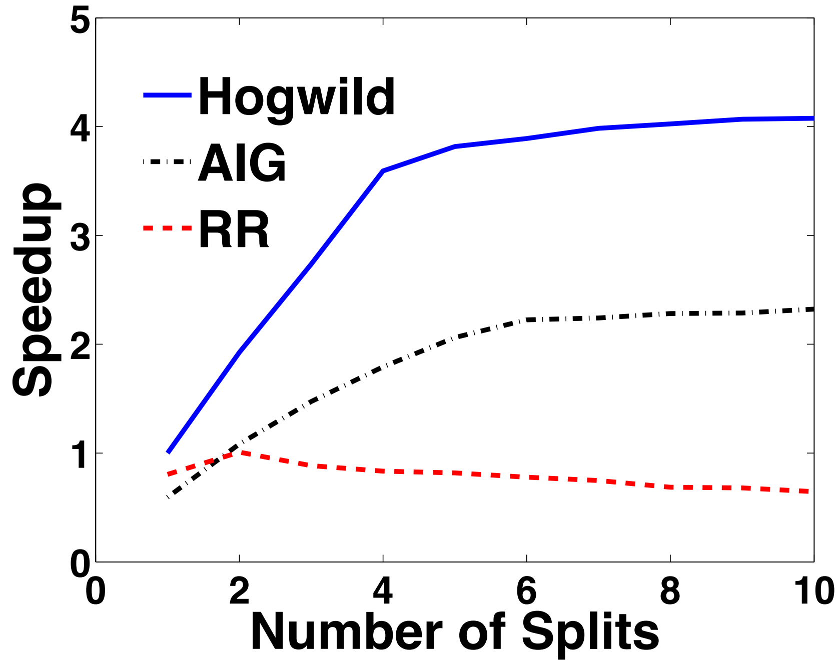
Speedups



SVM
RCV1



MC
Netflix



CUTS
Abdomen

Experiments run on 12 core machine
10 cores for gradients, 1 core for data shuffling

JELLYFISH

- SGD for Matrix Factorizations.

Example: $\text{minimize} \quad \sum_{(u,v)\in E}(X_{uv} - M_{uv})^2 + \mu\|\mathbf{X}\|_*$

- Idea: approximate $\mathbf{X} \approx \mathbf{L}\mathbf{R}^T$

$$\text{minimize}_{(\mathbf{L},\mathbf{R})} \sum_{(u,v)\in E} \{(\mathbf{L}_u\mathbf{R}_v^T - M_{uv})^2 + \mu_u\|\mathbf{L}_u\|_F^2 + \mu_v\|\mathbf{R}_v\|_F^2\}$$

- Step 1: Pick (i,j) and compute residual:

$$e = (\mathbf{L}_u\mathbf{R}_v^T - M_{uv})$$

- Step 2: Take a gradient step:

$$\begin{bmatrix} \mathbf{L}_u \\ \mathbf{R}_v \end{bmatrix} \leftarrow \begin{bmatrix} (1-\gamma\mu_u)\mathbf{L}_u - \gamma e\mathbf{R}_v \\ (1-\gamma\mu_v)\mathbf{R}_v - \gamma e\mathbf{L}_u \end{bmatrix}$$

JELLYFISH

Observation: With replacement sample=poor locality
Idea: Bias sample to improve locality.

$$\begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} \begin{bmatrix} R_1 & R_2 & R_3 \end{bmatrix} = \begin{bmatrix} L_1R_1 & L_1R_2 & L_1R_3 \\ L_2R_1 & L_2R_2 & L_2R_3 \\ L_3R_1 & L_3R_2 & L_3R_3 \end{bmatrix}$$

<u>Algorithm: Shuffle the data.</u>

1. Process $\{L_1R_1, L_2R_2, L_3R_3\}$ in parallel
2. Process $\{L_1R_2, L_2R_3, L_3R_1\}$ in parallel
3. Process $\{L_1R_3, L_2R_1, L_3R_2\}$ in parallel

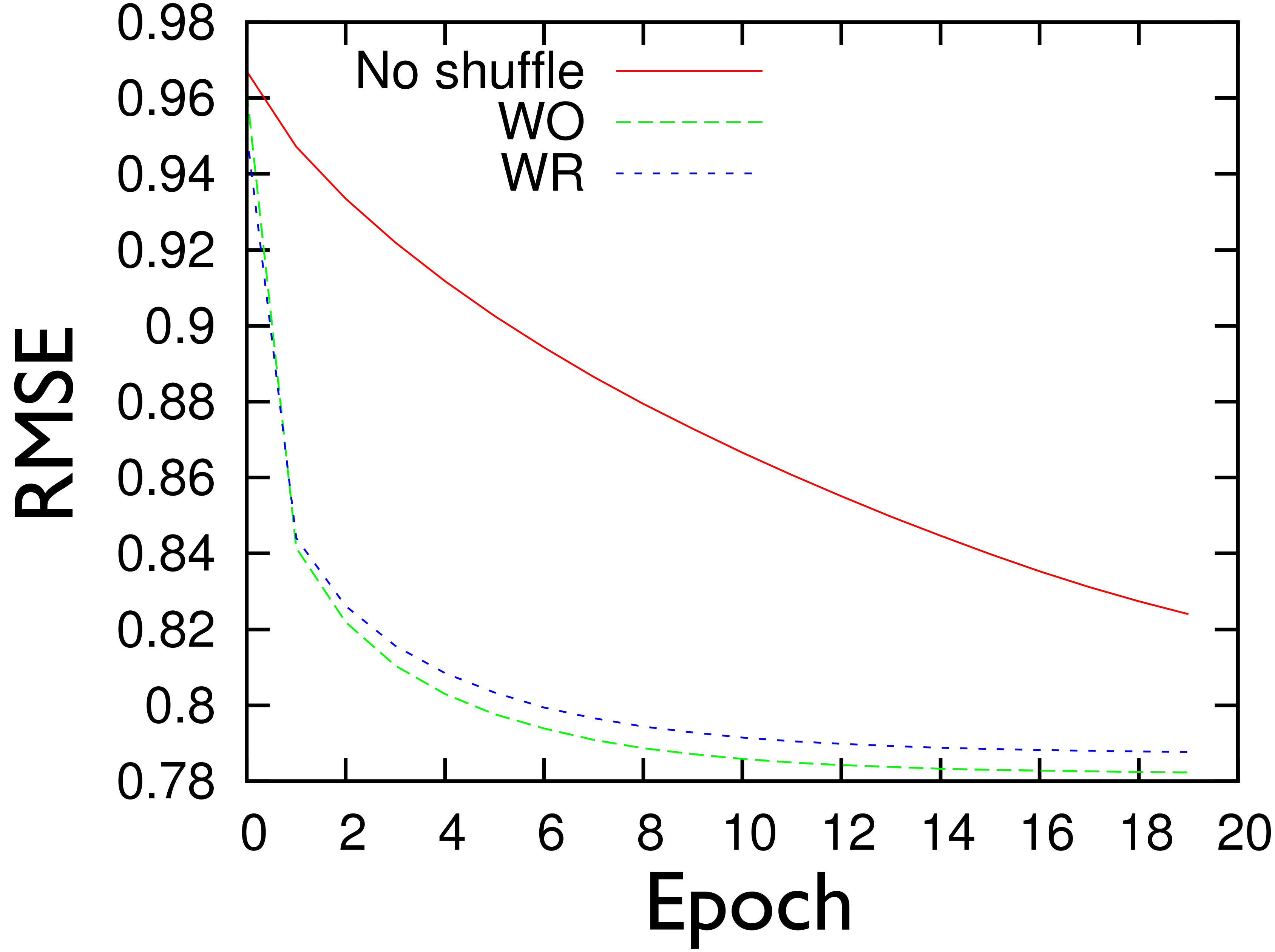
Big win: No locks!
(model access)

Recht and Ré, 2010

- 100x faster than standard solvers
- 25% speedup over HOGWILD! on 12 core machine
- 3x speedup on 40 core machine (sub minute timing)

- Extends to other structured matrix problems
  - max-norm
  - NMF (with non-negativity or simplex constraint)
    - LDA?

- Decoupling works for any algorithm where we can project the rows independently
- However, those not arising from SDPs have anything resembling guarantees.



SGD on Netflix Prize

- Theory: treats without replacement sampling like deterministic sampling.
- Practice: without replacement sampling *is always faster*.
- Open question: Why?

Simplest Problem? Least Squares

$$\text{minimize}_x \sum_{k=1}^{N} (a_k^T x - b_k)^2$$

Assume overdetermined: $a_k^T x_{\text{opt}} = b_k \qquad \forall k$

fixed order: $x_N = x_{\text{opt}} + \prod_{k=1}^{N} (I - \gamma a_k a_k^T)(x_0 - x_{\text{opt}})$

with replacement: $\mathbb{E}[x_N] = x_{\text{opt}} + \left(I - \frac{\gamma}{N} \sum_{k=1}^{N} a_k a_k^T \right)^N (x_0 - x_{\text{opt}})$

Which is better?

Example: Computing the mean

$$\text{minimize} \quad \sum_{k=1}^{4} (x-k)^2$$

Stepsize = 1/2k

$$x_0 = 0$$
$$x_1 = x_0 - (x_0 - 1) = 1$$
$$x_2 = x_1 - (x_1 - 2)/2 = 1.5$$
$$x_3 = x_2 - (x_2 - 3)/3 = 2$$
$$x_4 = x_3 - (x_3 - 4)/4 = 2.5$$

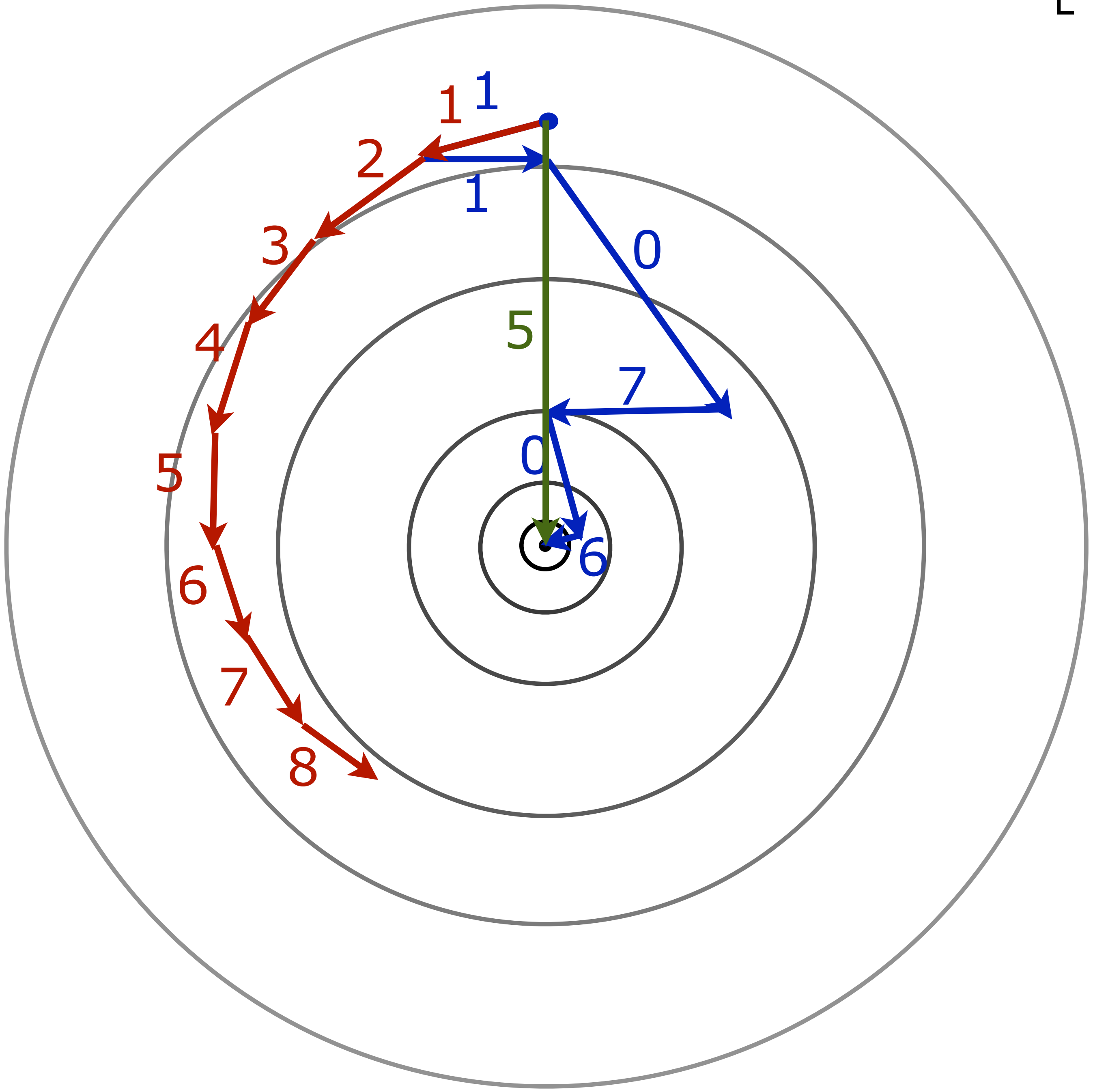
In general, if we minimize $\sum_{k=1}^{N} (x - z_k)^2$

SGD returns: $x_N = \frac{1}{N} \sum_{k=1}^{N} z_k$

$$\text{minimize} \quad \sum_{k=0}^{9} \left(\cos\left(\tfrac{\pi k}{10}\right) x_1 + \sin\left(\tfrac{\pi k}{10}\right) x_2\right)^2$$

Stepsize = 1/2

$$x - \frac{1}{2}\nabla f_j(x) = \frac{1}{2}\begin{bmatrix} 1 - c_j & -s_j \\ -s_j & 1 + c_j \end{bmatrix} x$$



Choose the best ordering

Choose a direction uniformly with replacement

Choose directions in order

Simple Question

- Given $A_1, \ldots, A_N \succeq 0$, DxD, is it true that
$$\left\| \prod_{i=1}^{N} A_i \right\| \leq \left\| \frac{1}{N} \sum_{i=1}^{N} A_i \right\|^N \quad ?$$
- If A_i are scalars, *this is always true* by the arithmetic-geometric mean inequality.
- Is it true for matrices?
- True for N=2 (Bhatia and Kittaneh 1990).

Simple Question

- Given $A_1, \ldots, A_N \succeq 0$, DxD, is it true that
$$\left\| \prod_{i=1}^{N} A_i \right\| \leq \left\| \frac{1}{N} \sum_{i=1}^{N} A_i \right\|^N \quad ?$$
- If A_i are scalars, *this is always true* by the arithmetic-geometric mean inequality.
- Is it true for matrices?
- True for N=2 (Bhatia and Kittaneh 1990).
- True for "generic'' A_i
- Straightforward bound:
$$\left\| \prod_{i=1}^{N} A_i \right\| \leq \left\| \frac{D}{N} \sum_{i=1}^{N} A_i \right\|^N$$

$$
\begin{aligned}
\left\| \prod_{i=1}^{N} A_i \right\| &\leq \prod_{i=1}^{N} \|A_i\| \\
&\leq \prod_{i=1}^{N} \operatorname{Tr}(A_i) \\
&\leq \left(\frac{1}{N} \sum_{i=1}^{N} \operatorname{Tr}(A_i) \right)^N \\
&= \operatorname{Tr} \left(\frac{1}{N} \sum_{i=1}^{N} A_i \right)^N \quad \leq \left\| \frac{D}{N} \sum_{i=1}^{N} A_i \right\|^N
\end{aligned}
$$

- Given $A_1, \ldots, A_N \succeq 0$, DxD, is it true that

$$\left\| \prod_{i=1}^{N} A_i \right\| \leq \left\| \frac{1}{N} \sum_{i=1}^{N} A_i \right\|^N$$

- No! Counterexample:

$$A_k = \begin{bmatrix} 1 + \cos\left(\frac{2\pi k}{N}\right) & \sin\left(\frac{2\pi k}{N}\right) \\ \sin\left(\frac{2\pi k}{N}\right) & 1 - \cos\left(\frac{2\pi k}{N}\right) \end{bmatrix}$$

$$\left\| \prod_{i=1}^{N} A_i \right\| = 2^N \cos^{N-1}\left(\frac{\pi}{N}\right) \approx 2^N \qquad \frac{1}{N} \sum_{i=1}^{N} A_i = I$$

- Saturates the bound

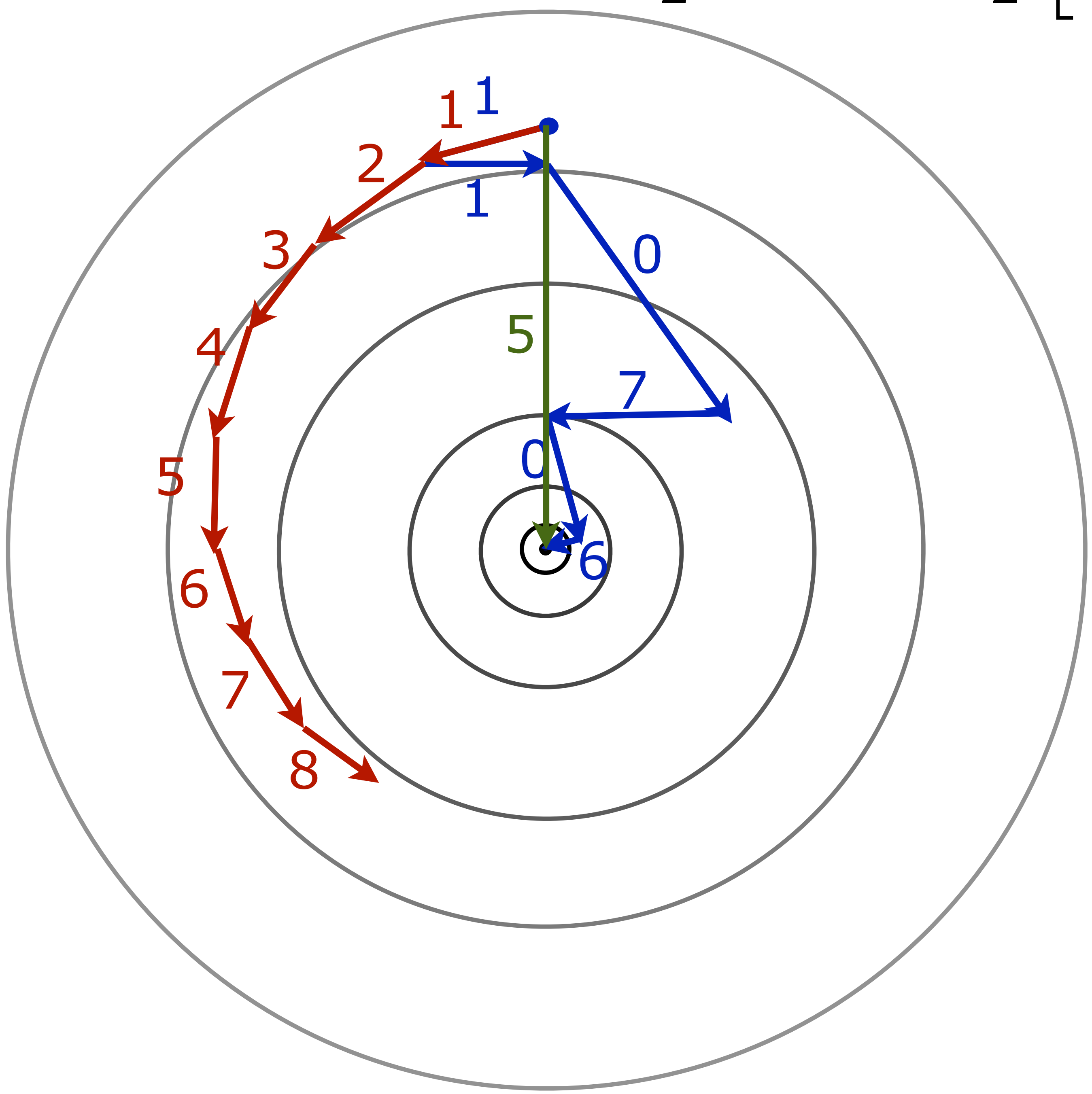
$$\left\| \prod_{i=1}^{N} A_i \right\| \leq \left\| \frac{D}{N} \sum_{i=1}^{N} A_i \right\|^N$$

- Similar counterexamples for all N,D>2

$$\text{minimize} \quad \sum_{k=0}^{9} \left(\cos\left(\tfrac{\pi k}{10}\right) x_1 + \sin\left(\tfrac{\pi k}{10}\right) x_2 \right)^2$$

Stepsize = 1/2

$$x - \frac{1}{2}\nabla f_j(x) = \frac{1}{2}\begin{bmatrix} 1 - c_j & -s_j \\ -s_j & 1 + c_j \end{bmatrix} x$$



Choose the best ordering

Choose a direction uniformly with replacement

Choose directions in order

What about *generically*?

- Given $A_1, \ldots, A_N \succeq 0$, DxD, is it true that

$$\left\| \prod_{i=1}^{N} A_i \right\| \leq \left\| \frac{1}{N} \sum_{i=1}^{N} A_i \right\|^N \quad ?$$

- If sampled from the normal distribution?

$$A_i = g_i g_i^* \qquad g_i \sim \mathcal{N}(0, \tfrac{1}{D} I_D)$$

$$\mathbb{E}\left[\prod_{i=1}^{N} A_i\right] = D^{-N} I$$

$$\mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^{N} A_i\right)^N\right] = Q_N$$

$$D^N \|Q_N\| \geq D^{N-1}\,\mathrm{Tr}(Q_N) = \left(1 + O(\tfrac{1}{N})\right) \sum_{k=0}^{N-1} \frac{(D/N)^k}{k+1} \binom{N}{k} \binom{N-1}{k}$$

$$\approx \left(1 + \sqrt{\frac{D}{N}}\right)^{2N}$$

$$A_i = g_i g_i^* \qquad g_i \sim \mathcal{N}(0, \tfrac{1}{D} I_D)$$

$$\boxed{\mathbb{E}\left[\prod_{i=1}^{N} A_i\right] = D^{-N} I}$$

$$\boxed{\mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^{N} A_i\right)^N\right] = Q_N}$$

$$D^N \|Q_N\| \geq D^{N-1}\,\mathrm{Tr}(Q_N)$$

$$= D^{N-1} N^{-N} \sum_{\{i_1,\ldots,i_N\}=1}^{D} \sum_{\{j_1,\ldots,j_N\}=1}^{N} \mathbb{E}[g_{i_1,j_1} g_{i_2,j_1} g_{i_2,j_2} g_{i_3,j_2} \cdots g_{i_N,j_N} g_{i_1,j_N}]$$

$$= D^{N} N^{-N} \sum_{\{i_2,\ldots,i_N\}=1}^{D} \sum_{\{j_1,\ldots,j_N\}=1}^{N} \mathbb{E}[g_{1,j_1} g_{i_2,j_1} g_{i_2,j_2} g_{i_3,j_2} \cdots g_{i_N,j_N} g_{1,j_N}]$$

$$\geq D^{N} N^{-N} \sum_{\{j_1,\ldots,j_N\}=1}^{N} \mathbb{E}[g_{1,j_1}^2 g_{1,j_2}^2 \cdots g_{1,j_N}^2]$$

$$\geq 1$$

But what about *on average*?

- Given $A_1, \ldots, A_N \succeq 0$, DxD, is it true that

$$\left\| \frac{1}{N!} \sum_{\sigma \in S_N} \prod_{i=1}^{N} A_{\sigma(i)} \right\| \leq \left\| \frac{1}{N} \sum_{i=1}^{N} A_i \right\|^N \quad ?$$

- Using the counterexample:

$$A_k = \begin{bmatrix} 1 + \cos\left(\frac{2\pi k}{N}\right) & \sin\left(\frac{2\pi k}{N}\right) \\ \sin\left(\frac{2\pi k}{N}\right) & 1 - \cos\left(\frac{2\pi k}{N}\right) \end{bmatrix}$$

$$\left\| \frac{1}{N!} \sum_{\sigma \in S_N} \prod_{i=1}^{N} A_{\sigma(i)} \right\| = -\,{}_2F_3 \left[\begin{matrix} 1 & -N/2 + 1/2 & -N/2 \\ 1/2 & -N+1 & \end{matrix} ; 1 \right]$$

$$\left\| \frac{1}{N} \sum_{i=1}^{N} A_i \right\| = \|I\| = 1$$

- Yet to find a counterexample for averaging!
- Is there a *noncommutative arithmetic-geometric mean inequality?*

Does the noncommutative arithmetic-geometric mean inequality hold?

- Given $A_1, \ldots, A_N \succeq 0$, DxD, is it true that

$$\left\| \frac{1}{N!} \sum_{\sigma \in S_N} \prod_{i=1}^{N} A_{\sigma(i)} \right\| \leq \left\| \frac{1}{N} \sum_{i=1}^{N} A_i \right\|^N \quad ?$$

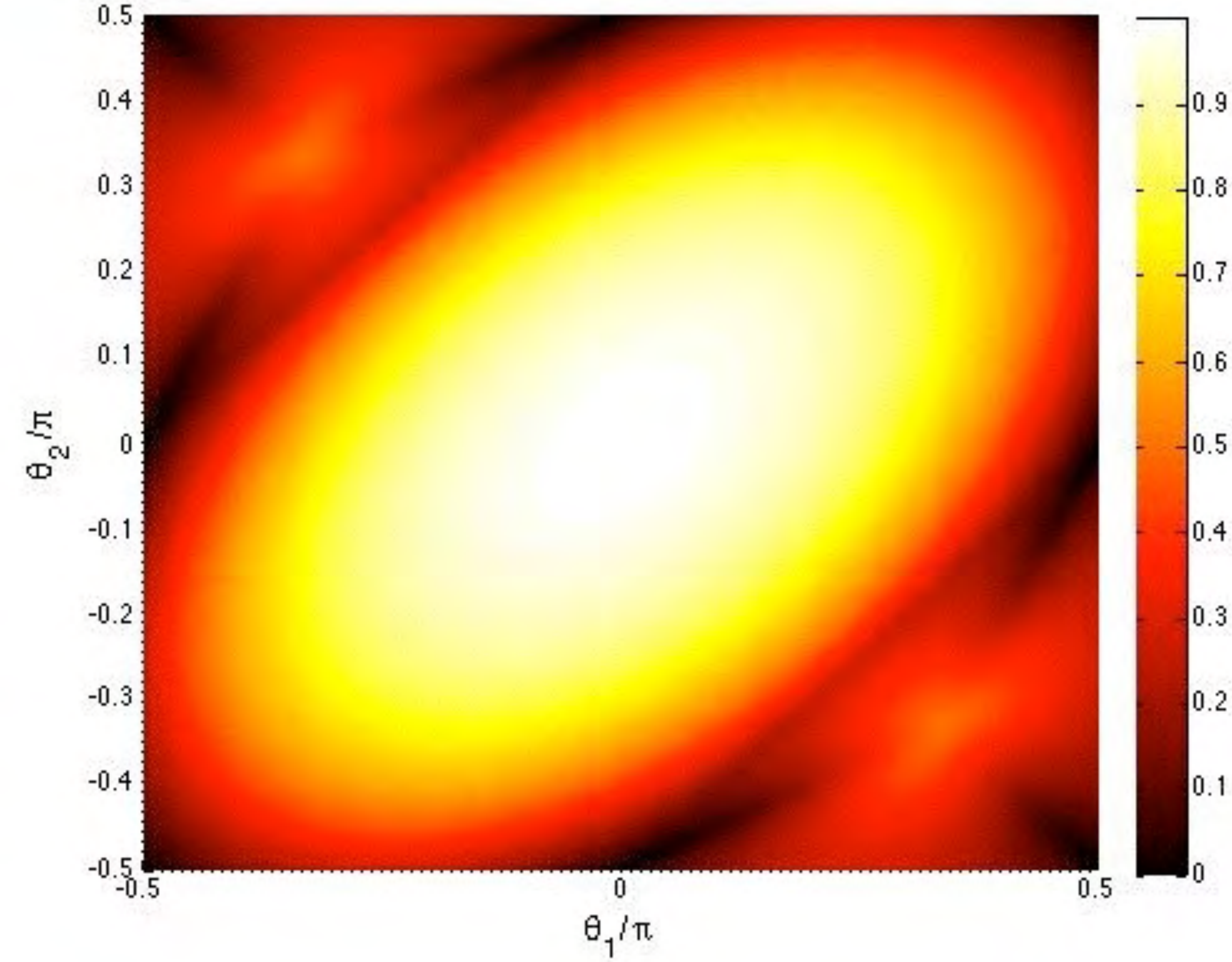
- Given $A_1, \ldots, A_N \succeq 0$, DxD, is it true that

$$\left\| \frac{1}{N!} \sum_{\sigma \in S_N} \prod_{i=1}^{K} A_{\sigma(i)} \right\| \leq \left\| \frac{1}{N} \sum_{i=1}^{N} A_i \right\|^K \quad ?$$

$$\left\| \frac{1}{N!} \sum_{\sigma \in S_N} \prod_{i=1}^{K} A_{\sigma(i)} \prod_{i=1}^{K} A_{\sigma(K-i+1)} \right\| \leq \left\| \frac{1}{N^k} \sum_{j_1, \ldots, j_k = 1}^{N} \prod_{i=1}^{K} A_{j_i} \prod_{i=1}^{K} A_{j_{k-i+1}} \right\|^N ?$$

- True for D=1
- True for N=2
- True for random matrices
- Does it hold in general?

$$\left\| \frac{1}{6} \sum_{\sigma \in S_3} \prod_{i=1}^{3} A_{\sigma(i)} \right\| / \left\| \frac{1}{3} \sum_{i=1}^{3} A_i \right\|^3$$



$$A_1 = \frac{1}{2} \begin{bmatrix} 1 - \cos(2\theta_1) & \sin(2\theta_1) \\ \sin(2\theta_1) & 1 + \cos(2\theta_1) \end{bmatrix}$$

$$A_2 = \frac{1}{2} \begin{bmatrix} 1 - \cos(2\theta_2) & \sin(2\theta_2) \\ \sin(2\theta_2) & 1 + \cos(2\theta_2) \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

The rank 1, 2x2, 3 matrix case

rank 1 asymptotically…

- Define the quantities (joint spectral radius)

$$\lambda_{\mathrm{wr}} = \lim_{k\to\infty} \frac{1}{k}\mathbb{E}_{\mathrm{wr}}\left[\log\left\|a_{i_k}a_{i_k}^T a_{i_{k-1}}a_{i_{k-1}}^T \cdots a_{i_1}a_{i_1}^T\right\|\right]$$

$$\lambda_{\mathrm{wo}} = \lim_{s\to\infty} \frac{1}{sn}\mathbb{E}_{\mathrm{wo}}\left[\log\left\|a_{i_{sn}}a_{i_{sn}}^T a_{i_{sn-1}}a_{i_{sn-1}}^T \cdots a_{i_1}a_{i_1}^T\right\|\right]$$

For rank one matrices:

$$\log\left\|a_{i_k}a_{i_k}^T a_{i_{k-1}}a_{i_{k-1}}^T \cdots a_{i_1}a_{i_1}^T\right\| = \log\|a_{i_k}\|_2 + \log\|a_{i_1}\|_2 + \sum_{j=1}^{k-1}\log|a_{i_j}^T a_{i_{j+1}}|$$

This immediately implies $\lambda_{\mathrm{wo}} \leq \lambda_{\mathrm{wr}}$

Specifically for SGD

$$\text{minimize}_x \sum_{k=1}^{n} (a_k^T x - b_k)^2$$

a_k random, iid

$$b_k = a_k^T x_\star + \omega_k$$

$$\mathbb{E}_{\text{wo}}[\|x_k - x_\star\|^2] = \mathbb{E}_{\text{wo}}[x_{k-1} - x_\star]^T (I - 2\gamma\Lambda + \gamma^2\Delta)\mathbb{E}_{\text{wo}}[x_{k-1} - x_\star] + \rho^2\gamma^2 \operatorname{trace}(\Lambda)$$

$$\mathbb{E}_{\text{wr}}[\|x_k - x_\star\|^2] = \mathbb{E}_{\text{wr}}\left[(x_{k-1} - x_\star)^T (I - 2\gamma\Lambda_n + \gamma^2\Delta_n)(x_{k-1} - x_\star)\right] + \rho^2\gamma^2 \operatorname{trace}(\Lambda)$$

$$\Lambda := \mathbb{E}[a_i a_i^T] \qquad \Delta := \mathbb{E}\left[\|a\|^2 a a^T\right]$$

$$\Lambda_n := \frac{1}{n}\sum_{i=1}^{n} a_i a_i^T \qquad \Delta_n := \frac{1}{n}\sum_{i=1}^{n} \|a_i\|^2 a_i a_i^T$$

Without replacement bound is tighter because sample averages are worse conditioned than expectations

Summary

- Practical Lessons
  - Don’t lock!
  - Locality Matters.

- Theory: Bias in sampling for better data access
  - understanding is only beginning here:
    - Noncommutative arithmetic-geometric mean in full generality
    - Biased orderings of SGD
    - Automatic identification of locality

Acknowledgements

- See:

  http://www.eecs.berkeley.edu/~brecht/publications.html

  for all references

- Reports:
  - HOGWILD!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent. Niu, Recht, Re, and Wright. 2011.
    - code: http://pages.cs.wisc.edu/~chrisre/hogwild.bz2
  - Parallel Stochastic Gradient Algorithms for Large-Scale Matrix Completion. Recht and Re. 2011.
    - code: http://pages.cs.wisc.edu/~chrisre/jellyfish.bz2
  - Beneath the Valley of the Noncommutative Arithmetic-Geometric Mean Inequality. Recht and Re. 2012.