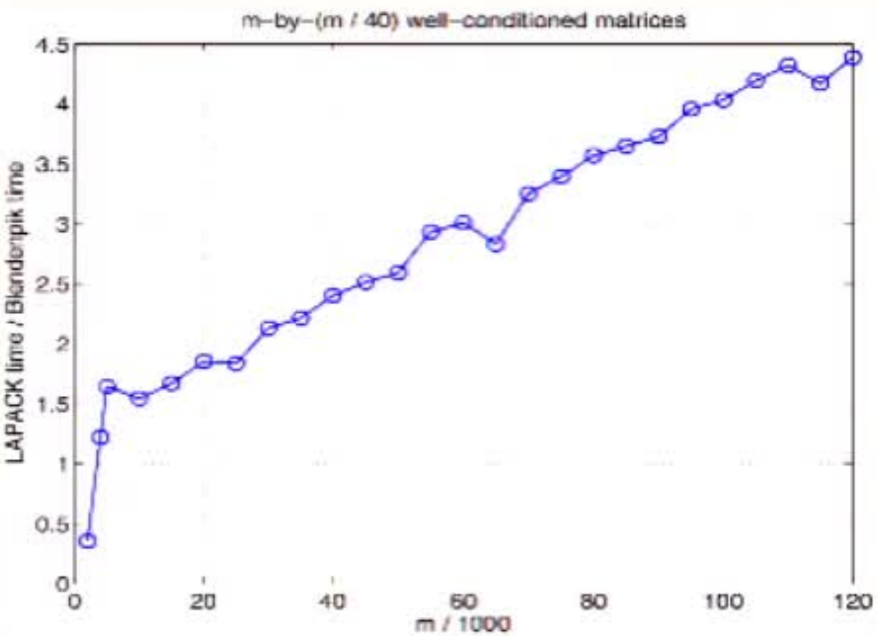
SKETCHING BASED MATRIX COMPUTATIONS FOR LARGE-SCALE DATA ANALYSIS

Haim Avron
Tel Aviv University
(work performed while at IBM Research)

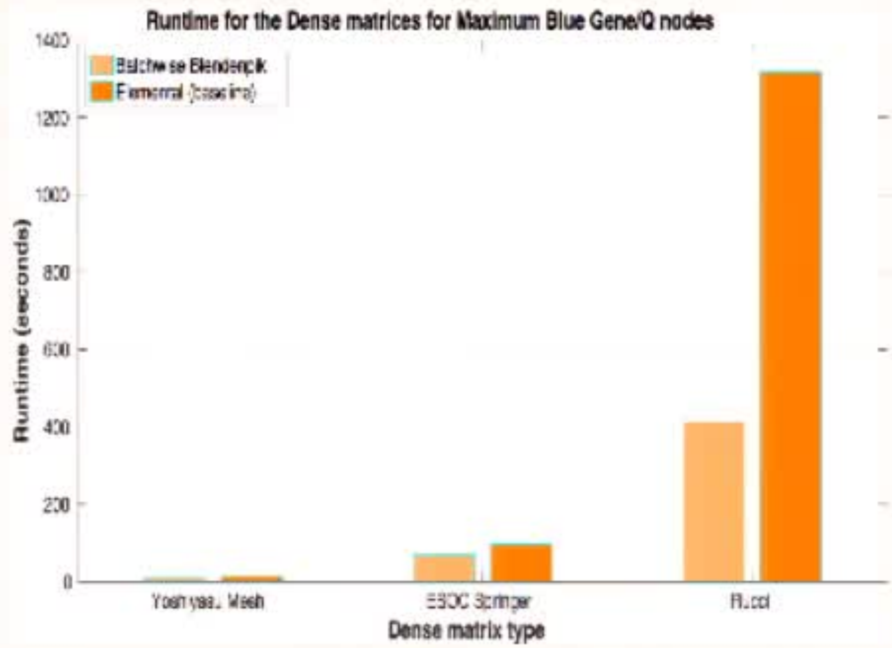
The Success of Sketching-based Linear Regression

Sequential on a laptop (2009):



(from Avron, Maymounkov and Toledo 2010)

Distributed-memory on BlueGene/Q (2015):



(from Chander et al. 2015)

Matrix Sketching

- A (randomized) transform that maintains some notion of geometry, e.g. Euclidean distance

$$\|\mathbf{Sx}\|_2 = (1 \pm \epsilon)\|\mathbf{x}\|_2,$$

on a subspace, e.g. for all $\mathbf{x} \in \mathcal{V}$ (with high probability).

- Two 'flavors' of use: "sketch-and-solve" and "sketch-to-precondition".

Sketch-and-solve:

- Use $\mathbf{S}$ to build a smaller problem, e.g.

$$\tilde{\mathbf{w}} = \operatorname{argmin}_{\mathbf{w}} \|\mathbf{SXw} - \mathbf{Sy}\|_2$$

- Problems:
  - *Not practical for high quality approximations*
  - *Might fail.*
- Advantages:
  - *Very fast for low quality approximations.*
  - *Usually good results for machine learning and data analysis applications.*

Sketch-to-precondition:

- Use $\mathbf{S}$ to precondition the problem, e.g.
  - *Factorize* $\mathbf{SX} = \mathbf{QR}$.
  - *Solve*

$$\mathbf{z} = \operatorname{argmin}_{\mathbf{z}} \left\|\mathbf{XR}^{-1}\mathbf{z} - \mathbf{y}\right\|_2.$$

  - *Return* $\mathbf{w} = \mathbf{R}^{-1}\mathbf{z}$.
- Advantages:
  - *Fast even for high quality approximations.*
  - *Failure results only in longer running times, and not in bad output.*

Warm-up: Linear Ridge Regression

(also called 'Tikhonov Regularization')

- Suppose $d \gg n$, and assume $\mathbf{X}$ is full rank.
- There are infinite solutions to $\mathbf{X}\mathbf{w} = \mathbf{y}$.
- It is common to add a "ridge regularizer" to make the solution unique

$$\mathbf{w} = \arg\min \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|_2^2$$

- Equivalent over-determined least squares:

$$\mathbf{w} = \arg\min \left\| \begin{bmatrix} \mathbf{X} \\ \sqrt{\lambda}\mathbf{I}_d \end{bmatrix} \mathbf{w} - \begin{bmatrix} \mathbf{y} \\ 0 \end{bmatrix} \right\|_2^2$$

- However $n + d \approx d$ so previously presented algorithms are not applicable.

Preconditioning by Sketching on the "Right"

- Rewriting the problem:

$$\mathbf{w} = \arg_{\mathbf{w}} \min \|\mathbf{w}\|_2^2 + \|\mathbf{z}\|_2^2 \quad s.t. \;\; \mathbf{X}\mathbf{w} + \sqrt{\lambda}\mathbf{z} = \mathbf{y}$$

- We we need to find the minimum norm solution for $\hat{\mathbf{X}}\hat{\mathbf{w}} = \mathbf{y}$

$$\hat{\mathbf{X}} = \begin{bmatrix}\mathbf{X} & \sqrt{\lambda}\mathbf{I}_n\end{bmatrix}$$

- "sketch on the right":
- *Sketch only* $\mathbf{X}$*, compute* $\mathbf{X}\mathbf{S}^{\mathbf{T}}$
- *Factorize* $\begin{bmatrix}\mathbf{X}\mathbf{S}^{\mathbf{T}} & \sqrt{\lambda}\mathbf{I}_n\end{bmatrix} = \mathbf{L}\mathbf{Q}$.
- *Use* $\mathbf{L}$ *as a preconditioner.*

- Remarks:
- *Sketching only* $\mathbf{X}$ *is motivated by the nonlinear case (later in the talk).*
- *Keeping the regularizer un-sketched costs very little (and we actually gain from it!).*

Analysis of Sketch-based Preconditioned Ridge Regression
(with Clarkson and Woodruff)

- An $\mathbf{S}$ "works" if for a fixed $\mathbf{A}$ and $\mathbf{B}$ and a selected c we have
$$\|\mathbf{A}^{\mathbf{T}}\mathbf{S}^{\mathbf{T}}\mathbf{S}\mathbf{B} - \mathbf{A}^{\mathbf{T}}\mathbf{B}\|_F \leq c\|\mathbf{A}\|_F\|\mathbf{B}\|_F$$

with high probability (aka probability of at least $1 - \delta$).

- Sketching dimension (number of rows in $\mathbf{S}$) depend on c, δ and #rows in $\mathbf{A}$ and $\mathbf{B}$.

- The relevant condition number is $\kappa\left(\mathbf{X}\mathbf{X}^{\mathbf{T}} + \lambda\mathbf{I}_n, \mathbf{X}\mathbf{S}^{\mathbf{T}}\mathbf{S}\mathbf{X}^{\mathbf{T}} + \lambda\mathbf{I}_n\right)$
- Suppose that $\mathbf{X} = \mathbf{L}_\lambda\mathbf{Q}_\lambda$ such that $\mathbf{L}_\lambda\mathbf{L}_\lambda^{\mathbf{T}} = \mathbf{X}\mathbf{X}^T + \lambda\mathbf{I}_n$. Then,

$$\kappa\left(\mathbf{X}\mathbf{X}^{\mathbf{T}} + \lambda\mathbf{I}_n, \mathbf{X}\mathbf{S}^{\mathbf{T}}\mathbf{S}\mathbf{X}^{\mathbf{T}} + \lambda\mathbf{I}_n\right) = \kappa\left(\mathbf{X}\mathbf{S}^{\mathbf{T}}\mathbf{S}\mathbf{X}^{\mathbf{T}} + \lambda\mathbf{I}_n, \mathbf{X}\mathbf{X}^{\mathbf{T}} + \lambda\mathbf{I}_n\right) = \kappa\left(\mathbf{X}\mathbf{S}^{\mathbf{T}}\mathbf{S}\mathbf{X}^{\mathbf{T}} + \lambda\mathbf{I}_n, \mathbf{L}_\lambda\mathbf{L}_\lambda^{\mathbf{T}}\right)$$
$$= \kappa\left(\mathbf{Q}_\lambda\mathbf{S}^{\mathbf{T}}\mathbf{S}\mathbf{Q}_\lambda^{\mathbf{T}} + \lambda\mathbf{L}_\lambda^{-1}\mathbf{L}_\lambda^{-\mathbf{T}}\right)$$

- To bound this we note that $\mathbf{Q}_\lambda\mathbf{Q}_\lambda^{\mathbf{T}} + \lambda\mathbf{L}_\lambda^{-1}\mathbf{L}_\lambda^{-\mathbf{T}} = \mathbf{I}_{\mathrm{n}}$, so

$$\|\mathbf{Q}_\lambda\mathbf{S}^{\mathbf{T}}\mathbf{S}\mathbf{Q}_\lambda^{\mathbf{T}} + \lambda\mathbf{L}_\lambda^{-1}\mathbf{L}_\lambda^{-\mathbf{T}} - \mathbf{I}_{\mathrm{n}}\|_F = \|\mathbf{Q}_\lambda\mathbf{S}^{\mathbf{T}}\mathbf{S}\mathbf{Q}_\lambda^{\mathbf{T}} - \mathbf{Q}_\lambda\mathbf{Q}_\lambda^{\mathbf{T}}\|_F$$

- So, select enough rows such that this term $\leq \frac{1}{2}$, which guarantees $\kappa \leq 3$.

Sparse Sketching (CountSketch)

- Defined by:
- *Random hash function* $h: \{1, \ldots, d\} \to \{1, \ldots, s\}$
- *Random sign function* $g: \{1, \ldots, d\} \to \{-1, +1\}$
- $(\mathbf{Sx})_i = \sum_{j \mid h(j)=i} g(j)\mathbf{x}_j$, *so* $\mathbf{Sx}$ *can be computed in* $O(\mathbf{nnz}(\mathbf{x}) + s)$.

Analysis of Sketch-based Preconditioned Ridge Regression
(with Clarkson and Woodruff)

- An $\mathbf{S}$ "works" if for a fixed $\mathbf{A}$ and $\mathbf{B}$ and a selected c we have

$$\left\|\mathbf{A}^\mathbf{T}\mathbf{S}^\mathbf{T}\mathbf{S}\mathbf{B} - \mathbf{A}^\mathbf{T}\mathbf{B}\right\|_F \leq c\|\mathbf{A}\|_F\|\mathbf{B}\|_F$$

with high probability (aka probability of at least $1 - \delta$).

- Sketching dimension (number of rows in $\mathbf{S}$) depend on c, δ and #rows in $\mathbf{A}$ and $\mathbf{B}$.

- The relevant condition number is $\kappa\left(\mathbf{X}\mathbf{X}^\mathbf{T} + \lambda\mathbf{I}_n, \mathbf{X}\mathbf{S}^\mathbf{T}\mathbf{S}\mathbf{X}^\mathbf{T} + \lambda\mathbf{I}_n\right)$
- Suppose that $\mathbf{X} = \mathbf{L}_\lambda\mathbf{Q}_\lambda$ such that $\mathbf{L}_\lambda\mathbf{L}_\lambda^\mathbf{T} = \mathbf{X}\mathbf{X}^T + \lambda\mathbf{I}_n$. Then,

$$\begin{aligned}\kappa\left(\mathbf{X}\mathbf{X}^\mathbf{T} + \lambda\mathbf{I}_n, \mathbf{X}\mathbf{S}^\mathbf{T}\mathbf{S}\mathbf{X}^\mathbf{T} + \lambda\mathbf{I}_n\right) = \kappa\left(\mathbf{X}\mathbf{S}^\mathbf{T}\mathbf{S}\mathbf{X}^\mathbf{T} + \lambda\mathbf{I}_n, \mathbf{X}\mathbf{X}^\mathbf{T} + \lambda\mathbf{I}_n\right) &= \kappa\left(\mathbf{X}\mathbf{S}^\mathbf{T}\mathbf{S}\mathbf{X}^\mathbf{T} + \lambda\mathbf{I}_n, \mathbf{L}_\lambda\mathbf{L}_\lambda^\mathbf{T}\right) \\ &= \kappa\left(\mathbf{Q}_\lambda\mathbf{S}^\mathbf{T}\mathbf{S}\mathbf{Q}_\lambda^\mathbf{T} + \lambda\mathbf{L}_\lambda^{-1}\mathbf{L}_\lambda^{-\mathbf{T}}\right)\end{aligned}$$

- To bound this we note that $\mathbf{Q}_\lambda\mathbf{Q}_\lambda^\mathbf{T} + \lambda\mathbf{L}_\lambda^{-1}\mathbf{L}_\lambda^{-\mathbf{T}} = \mathbf{I}_\mathrm{n}$, so

$$\left\|\mathbf{Q}_\lambda\mathbf{S}^\mathbf{T}\mathbf{S}\mathbf{Q}_\lambda^\mathbf{T} + \lambda\mathbf{L}_\lambda^{-1}\mathbf{L}_\lambda^{-\mathbf{T}} - \mathbf{I}_\mathrm{n}\right\|_F = \left\|\mathbf{Q}_\lambda\mathbf{S}^\mathbf{T}\mathbf{S}\mathbf{Q}_\lambda^\mathbf{T} - \mathbf{Q}_\lambda\mathbf{Q}_\lambda^\mathbf{T}\right\|_F$$

- So, select enough rows such that this term $\leq \frac{1}{2}$, which guarantees $\kappa \leq 3$.

Sparse Sketching (CountSketch)

- Defined by:
  - *Random hash function* $h: \{1, \ldots, d\} \to \{1, \ldots, s\}$
  - *Random sign function* $g: \{1, \ldots, d\} \to \{-1, +1\}$
  - $(\mathbf{Sx})_i = \sum_{j \mid h(j)=i} g(j)\mathbf{x}_j$, *so* $\mathbf{Sx}$ *can be computed in* $O(\mathbf{nnz}(\mathbf{x}) + s)$.

Alternative matrix definition:

- $\mathbf{S} = \mathbf{HD}$, where
  - $\mathbf{D}$ *is random diagonal, with* ± 1.
  - $\mathbf{H} \in \mathbb{R}^{s \times d}$ *has* H_{*j} *chosen randomly from* $\mathbf{e}_1, \ldots, \mathbf{e}_s$.

Lemma (Thorup and Zhang, 2012):

$$\Pr\left(\left\| \mathbf{A}^{\mathrm{T}}\mathbf{S}^{\mathrm{T}}\mathbf{S}\mathbf{B} - \mathbf{A}^{\mathrm{T}}\mathbf{B} \right\|_F \le \frac{3\sqrt{2}\|\mathbf{A}\|_F \|\mathbf{B}\|_F}{\sqrt{s\delta}} \right) \ge 1 - \delta$$

Sparse Sketching (CountSketch)

- Defined by:
- *Random hash function* $h: \{1, \dots, d\} \to \{1, \dots, s\}$
- *Random sign function* $g: \{1, \dots, d\} \to \{-1, +1\}$
- $(\mathbf{Sx})_i = \sum_{j \mid h(j)=i} g(j)\mathbf{x}_j$, *so* $\mathbf{Sx}$ *can be computed in* $O(\mathbf{nnz}(\mathbf{x}) + s)$.

Alternative matrix definition:

- $\mathbf{S} = \mathbf{HD}$, where
- $\mathbf{D}$ *is random diagonal, with* ± 1.
- $\mathbf{H} \in \mathbb{R}^{s \times d}$ *has* H_{*j} *chosen randomly from* $\mathbf{e_1}, \dots, \mathbf{e_s}$.

Sketch Size for Preconditioned Ridge Regression

- Recall:
  - *If* $\kappa\left(\mathbf{Q}_\lambda \mathbf{S}^\mathrm{T}\mathbf{S}\boldsymbol{Q}_\lambda^\mathrm{T} + \lambda \mathbf{L}_\lambda^{-1}\mathbf{L}_\lambda^{-\mathrm{T}}\right) \le 3$ *then we have a good preconditioner.*
  - *If* $\left\|\mathbf{Q}_\lambda \mathbf{S}^\mathrm{T}\mathbf{S}\boldsymbol{Q}_\lambda^\mathrm{T} - \mathbf{Q}_\lambda \mathbf{Q}_\lambda^\mathrm{T}\right\|_F \le \frac{1}{2}$, *then* $\kappa\left(\mathbf{Q}_\lambda \mathbf{S}^\mathrm{T}\mathbf{S}\boldsymbol{Q}_\lambda^\mathrm{T} + \lambda \mathbf{L}_\lambda^{-1}\mathbf{L}_\lambda^{-\mathrm{T}}\right) \le 3$.
- The last lemma ensures that with $s = O(\|\mathbf{Q}_\lambda\|_F^2)$ we have a good preconditioner.
- We have:

$$\mathbf{rank}_\lambda \equiv \|\mathbf{Q}_\lambda\|_F^2 = \sum_{i=1}^{n} \frac{\sigma_i^2}{\sigma_i^2 + \lambda}$$

- Always: $\mathbf{rank}_\lambda \le \boldsymbol{n}$, and can be much smaller (for large λ).
  (So: we benefit from not sketching the ridge term!)

$\mathbf{rank}_\lambda$ is known as the effective degrees of freedom in the statistics literature.

What about Sketch-and-Solve?

Chen et al. 2015:

- Compute $\tilde{\mathbf{w}} = \mathbf{X}^T\left(\mathbf{X}\mathbf{S}^{\mathbf{T}}\mathbf{S}\mathbf{X}^{\mathbf{T}} + \lambda\mathbf{I}_n\right)^{-1}\mathbf{y}$.
- With enough rows (depends on n), $\|\mathbf{w} - \tilde{\mathbf{w}}\|_2 \le \epsilon\|\mathbf{w}\|_2$.
- Doesn't work well when moving to nonlinear modeling...

(with Clarkson and Woodruff):

- Solve $\tilde{\mathbf{w}} = \arg\min\left\|\mathbf{X}\mathbf{S}^{\mathbf{T}}\mathbf{w} - \mathbf{y}\right\|_2^2 + \lambda\|\mathbf{w}\|_2^2$.
- With $O(\frac{\mathbf{rank}_\lambda^2}{\epsilon^2})$ rows in $\mathbf{S}$
  $(1-\epsilon)(\|\mathbf{X}\mathbf{w}-\mathbf{y}\|_2^2+\lambda\|\mathbf{w}\|_2^2) \le \left\|\mathbf{X}\mathbf{S}^{\mathbf{T}}\tilde{\mathbf{w}} - \mathbf{y}\right\|_2^2 + \lambda\|\tilde{\mathbf{w}}\|_2^2 \le (1+\epsilon)(\|\mathbf{X}\mathbf{w}-\mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_2^2)$.
- Rationale: at optimum, both objectives behave similarly.

Regularized Multivariate Polynomial Regression

- Let q be some degree parameter.
- We now try to fit a multivariate polynomial, i.e. $y \approx p_q(\mathbf{x})$.
- Interested in: biggish n (data size), small q (degree), moderate d (data dimension)
  - *E.g.* $n = 200{,}000,\ q = 3,\ \ d = 1000.$
  - *Assume* $d^q \gg n$.
- The problem is underdetermined, so we need to regularize:
  - *Let* $\mathbf{w}(p_q)$ *be a vector of* p_q*'s monomial coefficients. Use regularizer* $\lambda\|\mathbf{w}(p_q)\|_2^2$, i.e. solve

$$p_q = \arg\min_{p_q} \sum_{i=1}^{n} \left(p_q(\mathbf{x_i}) - y_i\right)^2 + \lambda\|\mathbf{w}(p_q)\|_2^2$$

  - *Remark: not clear if this is a good way to regularize the problem, but it is used in practice.*

As Linear Ridge Regression

- Define $V_q(\mathbf{X})$, a multivariate analogue of the Vandermonde matrix
- $V_q(\mathbf{X}) \in \mathbb{R}^{n \times (d+1)^q}$
- *Columns corresponds to monomials (a monomial may appear more than once).*
- *Rows corresponds to a data points.*
- *A row* $\mathbf{x}$ is mapped to $\phi([\mathbf{x}\ \mathbf{1}]) = [\mathbf{x}\ \mathbf{1}] \otimes \cdots \otimes [\mathbf{x}\ \mathbf{1}]$ (q times).
- *Example:*

$$V_2\left(\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix}\right) = \begin{bmatrix} 1 & x_{11} & x_{12} & x_{11} & x_{12} & x_{11}x_{12} & x_{12}x_{11} & x_{11}^2 & x_{12}^2 \\ 1 & x_{21} & x_{22} & x_{21} & x_{22} & x_{21}x_{22} & x_{22}x_{21} & x_{21}^2 & x_{22}^2 \end{bmatrix}$$

- Compute

$$\mathbf{w} = \arg\min \left\| V_q(\mathbf{X})\mathbf{w} - \mathbf{y} \right\|_2^2 + \lambda \|\mathbf{w}\|_2^2$$

and output is p_q such that $\mathbf{w}(p_q) = \mathbf{w}$. Specifically, $p_q(\mathbf{x}) = \phi([\mathbf{x}\ \mathbf{1}]) \cdot \mathbf{w}$.

Seems very expensive when d is not tiny.

Alternative Algorithm

- Observation 1: $\mathbf{w} = V_q(\mathbf{X})^{\mathrm{T}}\left(V_q(\mathbf{X})V_q(\mathbf{X})^{\mathrm{T}} + \lambda \mathrm{I}_n\right)^{-1}\mathbf{y}$
- Observation 2: $\left(V_q(\mathbf{X})V_q(\mathbf{X})^{\mathrm{T}}\right)_{\mathbf{ij}} = \left(\mathbf{x}_i \cdot \mathbf{x}_j + 1\right)^q$

- Efficient Multivariate Polynomial Regression:

- *Use observation 2 to compute* $V_q(\mathbf{X})V_q(\mathbf{X})^{\mathrm{T}}$ *in* $O(n^2 d \log q)$.
- *Compute* $\alpha = \left(V_q(\mathbf{X})V_q(\mathbf{X})^{\mathrm{T}} + \lambda \mathrm{I}_n\right)^{-1}\mathbf{y}$ *in* $O(n^3)$.
- *Via observation 1 we found polynomial* $p_q(\mathbf{x}) = \phi(\mathbf{x})V_q(\mathbf{X})^{\mathrm{T}}\alpha$.
- *Similar to observation 2,* $\phi(\mathbf{x})V_q(\mathbf{X})^{\mathrm{T}}$ *can be computed in* $O(nd \log q)$.
- *So,* $p_q(\mathbf{x})$ *can be computed in* $O(nd \log q)$.

Goal: accelerate this using sketching!

TensorSketch

(Pham and Pagh, 2013)

- $\mathbf{S} \in \mathbb{R}^{s\times(d+1)^q}$ defined by:
  - *q random hash functions* $h_1, \dots, h_q : \{1, \dots, d+1\} \to \{1, \dots, s\}$
  - *q random sign function* $g_1, \dots, g_q : \{1, \dots, d+1\} \to \{-1, +1\}$
  - *These define new sign and hash functions:*

$$H(i_1, \dots, i_q) = \sum_{j=1}^{q} h_j(i_j) \ \mathbf{mod}\ s$$

$$G(i_1, \dots, i_q) = \prod_{j=1}^{q} g_j(i_j)$$

  - $\mathbf{S}$ *is the CountSketch matrix defined by* H *and* G *(after indexing rows by tuples).*

TensorSketch can sometimes be applied quickly ($O(q(nnz(\mathbf{x}) + s\log s))$):

$$\phi(\mathbf{x})\mathbf{S}^{\mathrm{T}} = \mathbf{FFT}^{-1}\left(\mathbf{FFT}(\mathbf{x}\mathbf{S}_1^{\mathrm{T}}) \odot \cdots \odot \mathbf{FFT}(\mathbf{x}\mathbf{S}_q^{\mathrm{T}})\right)$$

where $\mathbf{S}_j$ is the CountSketch matrix defined by h_j and g_j.

Approximate Matrix Multiplication Properties

(with Nguyen and Woodruff)

Lemma:

Suppose $\mathbf{S}$ is a TensorSketch matrix with $s \geq \frac{2+3^q}{c^2\delta}$ rows, then

$$\Pr\left(\left\|\mathbf{A}^{\mathrm{T}}\mathbf{S}^{\mathrm{T}}\mathbf{S}\mathbf{B} - \mathbf{A}^{\mathrm{T}}\mathbf{B}\right\|_F \leq c\|\mathbf{A}\|_F\|\mathbf{B}\|_F\right) \geq 1 - \delta$$

Corollary:

$s = O(\mathbf{rank}_\lambda(V_q(\mathbf{X})^2))$ rows suffice for $V_q(\mathbf{X})\mathbf{S}^{\mathrm{T}}\mathbf{S}V_q(\mathbf{X})^{\mathrm{T}} + \lambda\mathrm{I}_n$ to be a good preconditioner for $V_q(\mathbf{X})V_q(\mathbf{X})^{\mathrm{T}} + \lambda\mathrm{I}_n$.

Efficient Use of the Preconditioner

- The preconditioner is only useful if $s < n$.
- *Otherwise, we might as well compute and factor* $V_q(\mathbf{X})V_q(\mathbf{X})^{\mathrm{T}} + \lambda \mathrm{I}_n$.

- For $s < n$, we can do better than computing and factoring $V_q(\mathbf{X})\mathbf{S}^{\mathrm{T}}\mathbf{S}V_q(\mathbf{X})^{\mathrm{T}} + \lambda \mathrm{I}_n$.

- The Woodbury matrix identity imply that
$\left(V_q(\mathbf{X})\mathbf{S}^{\mathrm{T}}\mathbf{S}V_q(\mathbf{X})^{\mathrm{T}} + \lambda \mathrm{I}_n\right)^{-1} = \lambda^{-1}\left(\mathrm{I}_n - V_q(\mathbf{X})\mathbf{S}^{\mathrm{T}}\left(\mathbf{S}V_q(\mathbf{X})^{\mathrm{T}}V_q(\mathbf{X})\mathbf{S}^{\mathrm{T}} + \boldsymbol{\lambda}\mathbf{I}_s\right)^{-1}\mathbf{S}V_q(\mathbf{X})^{\mathrm{T}}\right)$

- So, with $O(ns^2)$ preprocessing we can apply the preconditioner efficiently.

Faster Regularized Multivariate Polynomial Regression

1. Compute $\mathbf{K} = V_q(\mathbf{X})V_q(\mathbf{X})^{\mathrm{T}} + \lambda\mathbf{I}_n$ (cost: $O(n^2 d \log q)$)

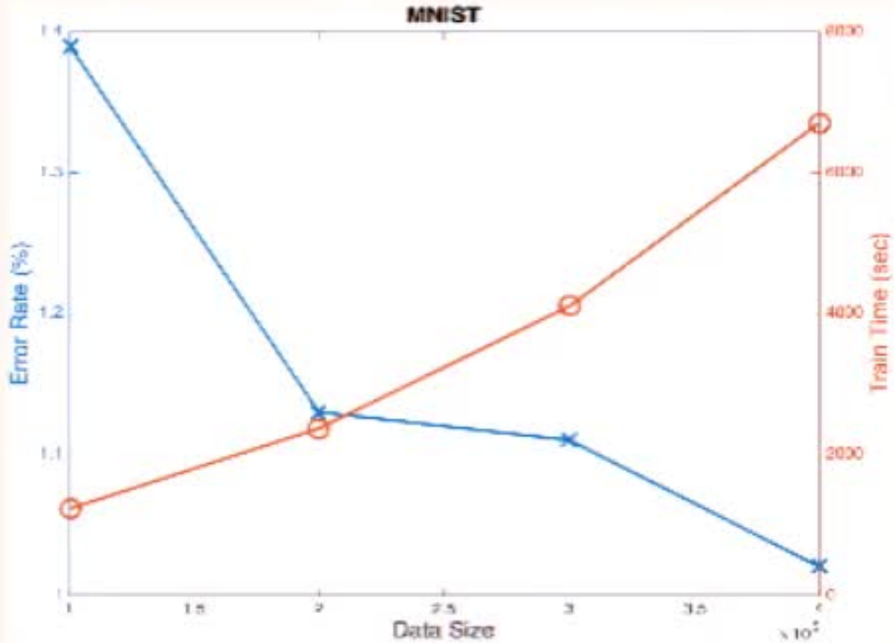
2. Compute $\mathbf{Z} = V_q(\mathbf{X})\mathbf{S}^T$ (cost: $O(q(\mathrm{nnz}(\mathbf{X}) + s \log s))$)

3. Factorize $\begin{bmatrix} \mathbf{Z} \\ \sqrt{\lambda}\mathbf{I}_s \end{bmatrix} = \mathbf{QR}$ (cost: $O(ns^2)$)

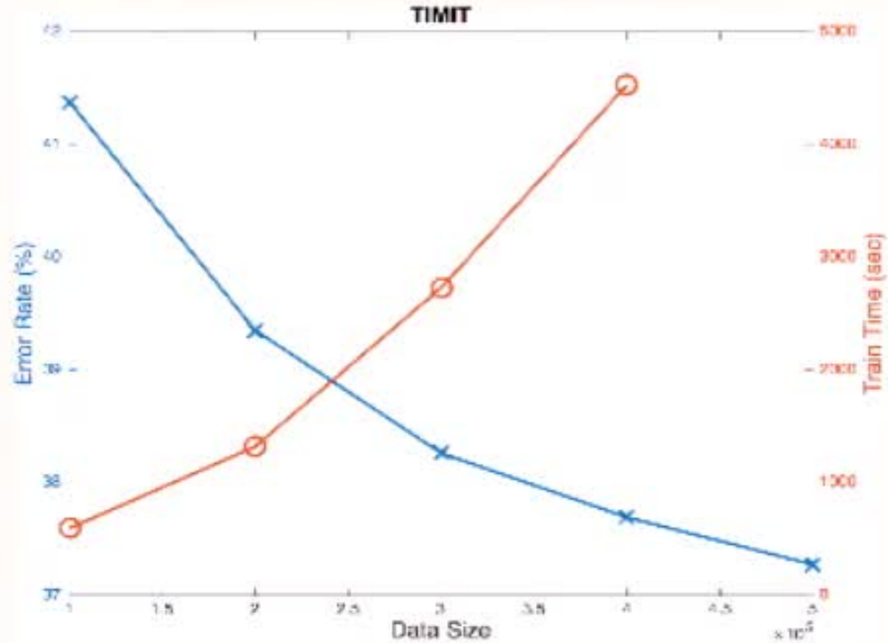
4. Use CG to solve $\mathbf{K\alpha} = \mathbf{y}$ using $\lambda^{-1}(\mathbf{I}_n - \mathbf{Z}\mathbf{R}^{-1}\mathbf{R}^{-\mathrm{T}}\mathbf{Z}^{\mathrm{T}})$ as preconditioner (**cost**: $O(n^2)$ per iteration)

Larege Scale Multivariate Polynomial Regression

Dataset from an image processing application:



Dataset from a speech recognition application:



- Classification problems; solved via regression using standard techniques.
- Degree of polynomial is 3 for MNIST and 4 for TIMIT. For both datasets we rescale the features.
- Run on BlueGene/Q using 128 nodes (= 2,048 cores).

Sketching for the Gaussian Kernel: Random Fourier Features

(Rahimi and Recht, 2007)

- Observation (due to Bochener's Theorem):

$$k(\mathbf{x}, \mathbf{z}) = \mathrm{E}_{\mathbf{w}}(\exp(-i\mathbf{w}^T(\mathbf{x} - \mathbf{z})), \qquad \mathbf{w} \sim N(0, \sigma^{-2}\mathbf{I}_d)$$

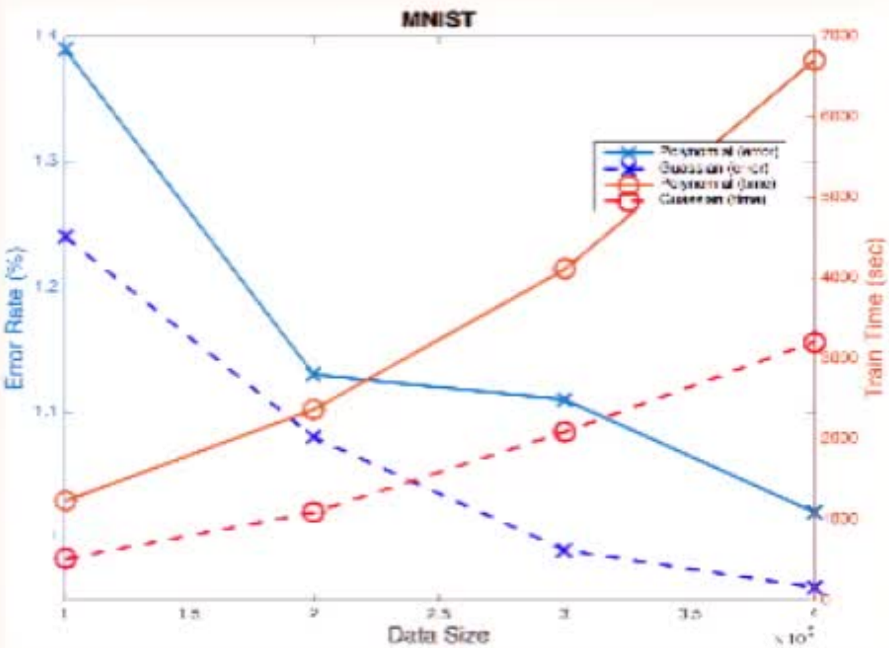
- The sketch: sample $\mathbf{w}_1, \ldots, \mathbf{w}_s$ and map

$$\mathbf{x} \to \frac{1}{\sqrt{s}}[e^{-i\mathbf{w}_1^T\mathbf{x}} \quad \ldots \quad e^{-i\mathbf{w}_s^T\mathbf{x}}].$$

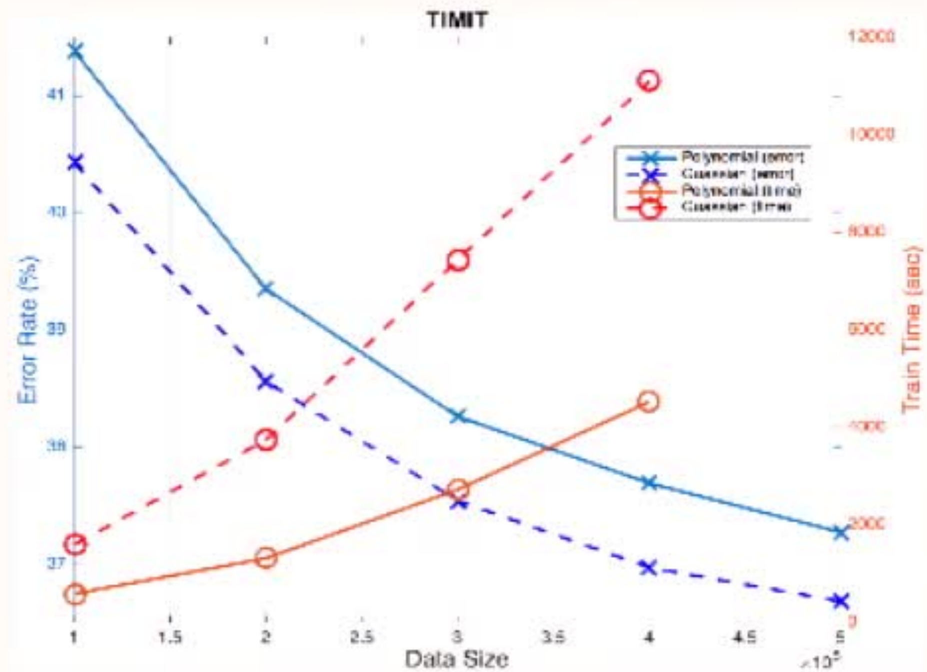
- There is no proof that this sketch has strong matrix multiplication guarantees.

Preconditioned Solver Works Well

Dataset from an image processing application:



Dataset from a speech recognition application:



- Classification problems; solved via regression using standard techniques.
- Degree of polynomial is 3 for MNIST and 4 for TIMIT. For both datasets we rescale the features.
- Run on BlueGene/Q using 128 nodes (= 2,048 cores).

Conclusions

- Matrix sketching is a powerful technique for designing new exciting algorithms.
- So far, it mostly addressed problems motivated by linear modeling.
- However, effectively leveraging "big data" requires nonlinear and nonparametric modeling.
- Matrix sketching can help for nonlinear modeling as well, but there is still much to be done.

Acknowledgements

- Work was done while at IBM Research.

- Collaborators:
  - *Ken Clarkson, Christopher Carothers, Petros Drineas, Po-sen Huang, Yves Ineichen, Chander Iyer, Georgios Kollias, Petar Maymounkov, Huy Nguyen, Bhuvana Ramabhardan, Tara Sainath, Vikas Sindhwani, Sivan Toledo, David Woodruff*

- Support of the XDATA program of the Defense Advanced Research Projects Agency (DARPA), administered through Air Force Research Laboratory contract FA8750-12-C-0323.

More General: Kernel Ridge Regression

- Multivariate polynomial regression is a special case of kernel ridge regression.
- In kernel ridge regression we start with a kernel $k: \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$.
- The kernel defines an Hilbert space $\mathcal{H}$.
- We search for functions in $\mathcal{H}$, i.e. solve

$$\arg\min_{f \in \mathcal{H}} \sum_{i=1}^{n} \left(y_i - f(\mathbf{x}_i)\right)^2 + \lambda \|f\|_{\mathcal{H}}^2.$$

- Skipping ... some ... mathematical ... details, the solution is

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x})$$

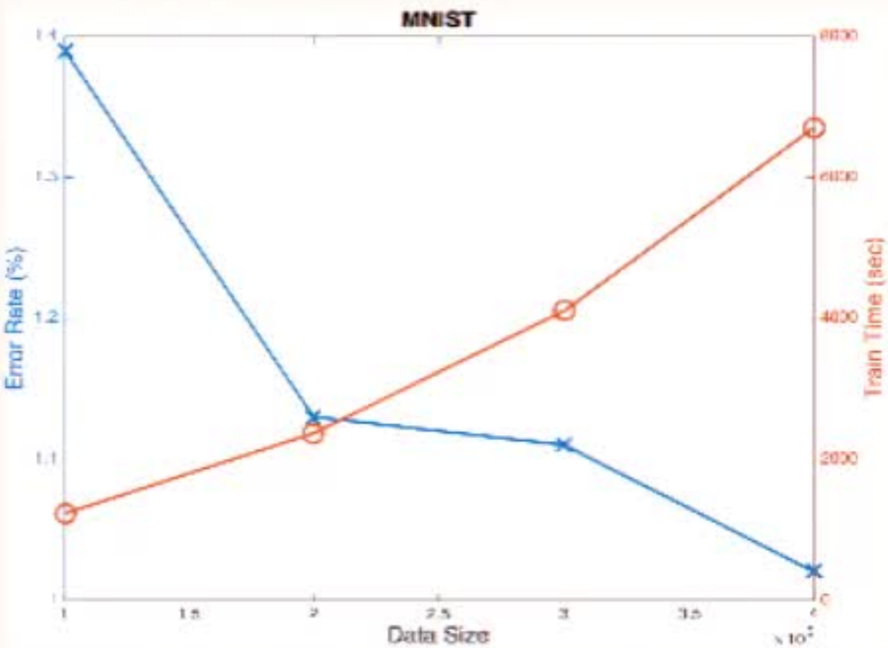
where

$$(\mathbf{K} + \lambda \mathbf{I}_n)\boldsymbol{\alpha} = \mathbf{y}$$

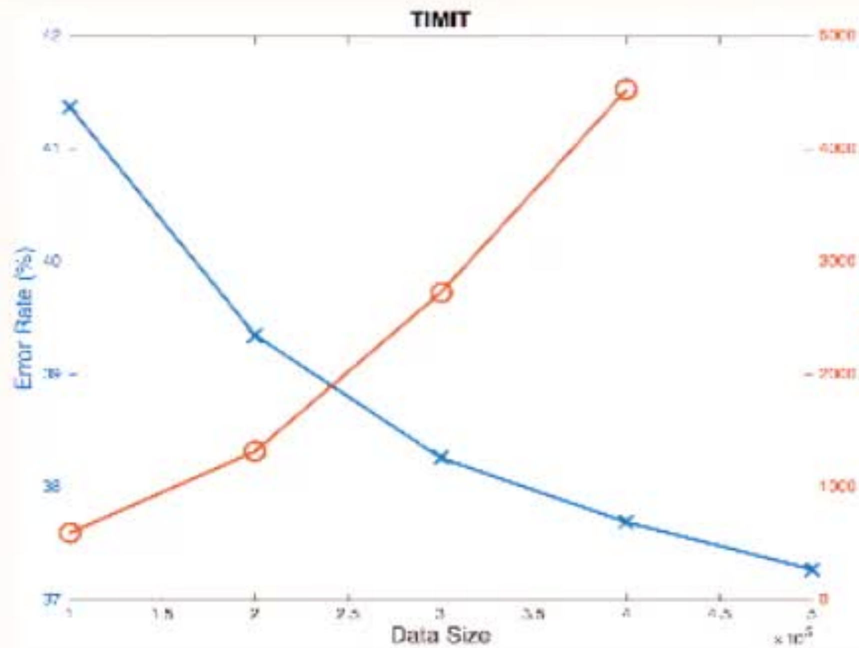
with $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

Larege Scale Multivariate Polynomial Regression

Dataset from an image processing application:



Dataset from a speech recognition application:



- Classification problems: solved via regression using standard techniques.
- Degree of polynomial is 3 for MNIST and 4 for TIMIT. For both datasets we rescale the features.
- Run on BlueGene/Q using 128 nodes (= 2,048 cores).

TENSORSKETCH
(Pham and Pagh, 2013)

- $\mathbf{S} \in \mathbb{R}^{s \times (d+1)^q}$ defined by:
  - *q random hash functions* $h_1, \ldots, h_q : \{1, \ldots, d+1\} \to \{1, \ldots, s\}$
  - *q random sign function* $g_1, \ldots, g_q : \{1, \ldots, d+1\} \to \{-1, +1\}$
  - *These define new sign and hash functions:*

$$H(i_1, \ldots, i_q) = \sum_{j=1}^{q} h_j(i_j) \ \mathbf{mod}\ s$$

$$G(i_1, \ldots, i_q) = \prod_{j=1}^{q} g_j(i_j)$$

  - $\mathbf{S}$ *is the* COUNTSKETCH *matrix defined by* H *and* G *(after indexing rows by tuples).*

TENSORSKETCH can sometimes be applied quickly $(O(q(nnz(\mathbf{x}) + s \log s)))$:

$$\phi(\mathbf{x})\mathbf{S}^{\mathrm{T}} = \mathbf{FFT}^{-1}\left(\mathbf{FFT}(\mathbf{x}\mathbf{S}_1^{\mathrm{T}}) \odot \cdots \odot \mathbf{FFT}(\mathbf{x}\mathbf{S}_q^{\mathrm{T}})\right)$$

where $\mathbf{S}_j$ is the COUNTSKETCH matrix defined by h_j and g_j.

Efficient Use of the Preconditioner

- The preconditioner is only useful if $s < n$.
  - *Otherwise, we might as well compute and factor* $V_q(\mathbf{X})V_q(\mathbf{X})^{\mathrm{T}} + \lambda \mathrm{I}_n$.
- For $s < n$, we can do better than computing and factoring $V_q(\mathbf{X})\mathbf{S}^{\mathrm{T}}\mathbf{S}V_q(\mathbf{X})^{\mathrm{T}} + \lambda \mathrm{I}_n$.
- The Woodbury matrix identity imply that
  $\left(V_q(\mathbf{X})\mathbf{S}^{\mathrm{T}}\mathbf{S}V_q(\mathbf{X})^{\mathrm{T}} + \lambda \mathrm{I}_n\right)^{-1} = \lambda^{-1}\left(\mathrm{I}_n - V_q(\mathbf{X})\mathbf{S}^{\mathrm{T}}\left(\mathbf{S}V_q(\mathbf{X})^{\mathrm{T}}V_q(\mathbf{X})\mathbf{S}^{\mathrm{T}} + \lambda \mathbf{I}_s\right)^{-1}\mathbf{S}V_q(\mathbf{X})^{\mathrm{T}}\right)$
- So, with $O(ns^2)$ preprocessing we can apply the preconditioner efficiently.

Approximate Matrix Multiplication Properties

(with Nguyen and Woodruff)

Lemma:

Suppose $\mathbf{S}$ is a TensorSketch matrix with $s \geq \frac{2+3^q}{c^2\delta}$ rows, then

$$\Pr\left(\left\|\mathbf{A}^{\mathrm{T}}\mathbf{S}^{\mathrm{T}}\mathbf{S}\mathbf{B} - \mathbf{A}^{\mathrm{T}}\mathbf{B}\right\|_F \leq c\|\mathbf{A}\|_F\|\mathbf{B}\|_F\right) \geq 1 - \delta$$

Corollary:

$s = O(\mathbf{rank}_\lambda(V_q(\mathbf{X})^2))$ rows suffice for $V_q(\mathbf{X})\mathbf{S}^{\mathrm{T}}\mathbf{S}V_q(\mathbf{X})^{\mathrm{T}} + \lambda\mathrm{I}_n$ to be a good preconditioner for $V_q(\mathbf{X})V_q(\mathbf{X})^{\mathrm{T}} + \lambda\mathrm{I}_n$.