SketchySVD

Joel A. Tropp

Computing + Mathematical Sciences

California Institute of Technology

jtropp@cms.caltech.edu

Coauthors: Alp Yurtsever (EPFL), Volkan Cevher (EPFL);

Yiming Sun (Cornell), Yang Guo (UWisc), Charlene Luo (Columbia), Madeleine Udell (Cornell)

Thanks: Gunnar Martinsson (UT-Austin), Mark Tygert (Facebook)

Research supported by ONR, AFOSR, NSF, DARPA, Sloan, and Moore.

Open for Business

SIAM Journal on Mathematics of Data Science (SIMODS)

https://simods.siam.org/

Editors: Tammy Kolda (EIC); Al Hero, Mike Jordan, Rob Nowak, Joel A. Tropp

Building a Community

First SIAM Conference on Mathematics of Data Science (MDS 2020)

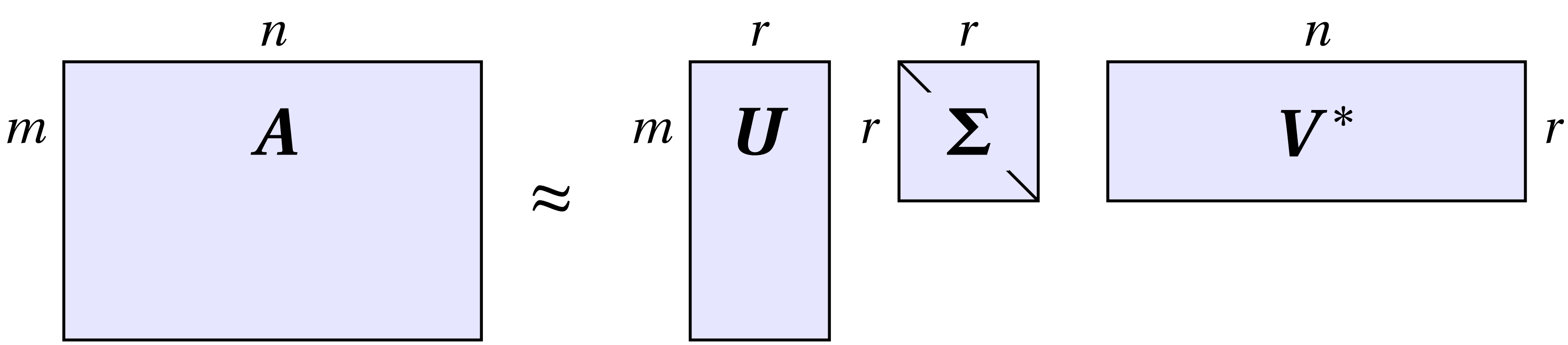
5–7 May 2020

Cincinnati, Ohio, USA

Co-Chairs: Gitta Kutyniok, Ali Pinar, Joel A. Tropp

The Famous Truncated SVD

Truncated Singular Value Decomposition (TSVD)



- $\boldsymbol{U}, \boldsymbol{V}$ have orthonormal columns and $\boldsymbol{\Sigma}$ is nonnegative diagonal
- **Interpretation:** r-truncated SVD = optimal rank-r approximation
- Approximately $r(m+n)$ degrees of freedom

Applications:

- Least-squares computations (linear regression)
- Principal component analysis (orthogonal regression; total least squares)
- Summarization, data reduction, visualization, ...

A Pæan to the Truncated SVD

"[Truncated SVD] is one of the few methods that has solid foundations and can be trusted, provided that the computations are correct. Having something reliable in machine learning is worth its weight in gold — there are almost no gold standards in the field, precluding rapid progress. Think of what happens to machine learning when the routine for matrix–vector multiplication doesn't always work right. You can't debug any code, much less a big system consisting of many algorithms thrown together."

–Nemo

Modern Numerical Linear Algebra

What's Wrong with Classical TSVD Algorithms?

- Nothing... when the matrices are small and fit in core memory

Climate Change:

- Medium- to large-scale data (Gigabytes+)
- New architectures (multi-core, distributed, data centers, ...)
- **Today:** New data presentations (dynamic, off-core, streaming)

Engineering:

- Theoretically, we already know how to do streaming TSVD, but...
- Many current algorithms are not ready for implementation
- For scientific applications, high accuracy is essential!
- **Today:** First practical algorithms for streaming TSVD with high accuracy

History of Randomized SVD Algorithms

Dimension Reduction:

- Gaussian maps (Johnson & Lindenstrauss 1984; Indyk & Motwani 1998)
- **Sparse maps** (Achlioptas 2001; Charikar et al. 2002; Clarkson & Woodruff 2011; ...)
- Randomized Fourier transforms (Ailon & Chazelle 2006; Woolfe et al. 2008)
- **Tensor random projections** (Rudelson 2011; Sun et al. 2018)

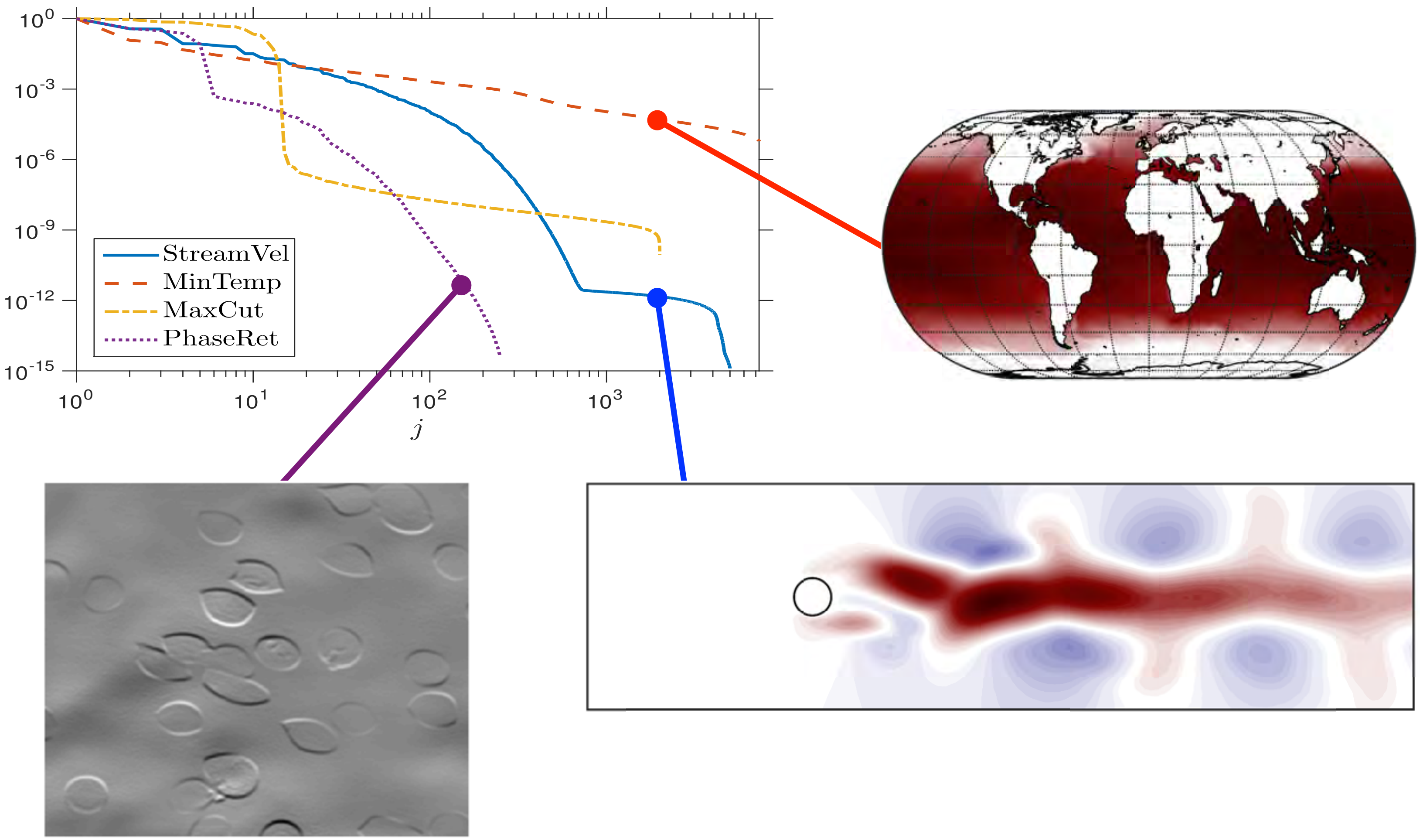
Theory:

- Randomized algorithms for LSI (Frieze, Kannan, Papadimitriou, Vempala 1998)
- Randomized linear algebra foundations (Drineas, Kannan, Mahoney 2004)
- Sketch-and-solve framework (Sarlós 2006)
- Streaming linear algebra foundations (Clarkson & Woodruff 2009)
- **Streaming SVD from three sketches** (Upadhyay 2016)

Practice:

- Low-rank matrix approximation algorithms (Martinsson, Rokhlin, Tygert 2004)
- Fast, one-pass matrix approximation (Woolfe, Liberty, Rokhlin, Tygert 2008)
- **Randomized SVD framework, algorithms, and analysis** (Halko, Martinsson, Tropp 2009)
- Randomized block Krylov methods (Halko, Martinsson, Szlam, Tygert 2011)
- **Practical streaming SVD algorithms** (Tropp, Yurtsever, Udell, Cevher 2016–2019)

Spectral Decay in Scientific Data



Streaming Linear Algebra

The Turnstile Model:

$$\boldsymbol{A} \quad = \quad \boldsymbol{H}_1 + \boldsymbol{H}_2 + \boldsymbol{H}_3 + \boldsymbol{H}_4 + \cdots$$

- Huge input matrix $\boldsymbol{A}$ is presented as a sum of innovations $\boldsymbol{H}_i$
- Must discard each innovation $\boldsymbol{H}_i$ after it is processed
- **Goal:** Without storing $\boldsymbol{A}$ in full, return TSVD after seeing all updates

Applications:

- One-pass approximation of matrix stored out-of core
- Large-scale semidefinite programming algorithms
- Scientific simulation and data collection

Sources: Muthukrishnan 2008; Woolfe et al. 2008; Clarkson & Woodruff 2009; HMT 2011; Woodruff 2014; TYUC 2016–2019; ....

Randomized Linear Sketches

$$\text{sketch} \quad = \quad \mathcal{L}(\boldsymbol{A}) \quad = \quad \sum_i \mathcal{L}(\boldsymbol{H}_i)$$

- Select a **linear** map $\mathcal{L}$ without reference to $\boldsymbol{A}$
- Sketch dimension is much smaller than input matrix dimension
- Use **randomness** so sketch works for an arbitrary input
- **[LNW14] Essentially the only way to handle the turnstile model!**

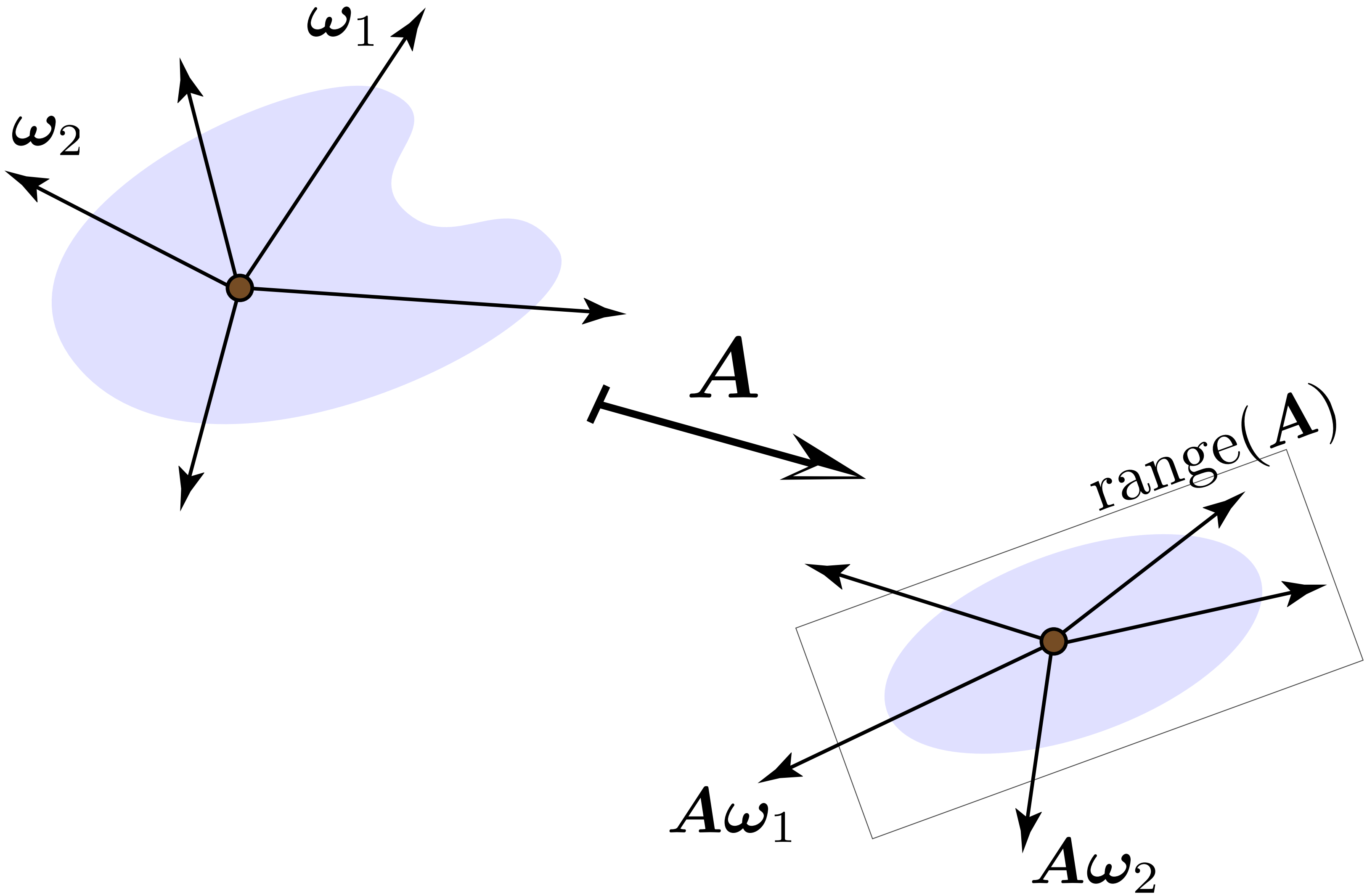
Examples:

- Left multiply: $\mathcal{L}(\boldsymbol{A}) = \boldsymbol{\Upsilon}\boldsymbol{A}$ where $\boldsymbol{\Upsilon}$ is fat
- Right multiply: $\mathcal{L}(\boldsymbol{A}) = \boldsymbol{A}\boldsymbol{\Omega}$ where $\boldsymbol{\Omega}$ is tall
- Select some entries: $\mathcal{L}(\boldsymbol{A}) = \{a_{ij} : (i,j) \in E\}$

Sources: Alon et al. 1996; Sarlós 2006; Muthukrishnan 2008; Woolfe et al. 2008; Clarkson & Woodruff 2009; HMT 2011; Mahoney 2012; Woodruff 2014; Li et al. 2014; Drineas & Mahoney 2016; TYUC 2016–2019; ....

The Randomized Range Finder

Random Vectors Align with the Range



The Randomized Range Finder

Q. How do we use a randomized linear sketch to approximate the range of $\boldsymbol{A}$?

A. Just multiply random vectors into $\boldsymbol{A}$ and orthogonalize!

1. Draw a random matrix $\boldsymbol{\Omega}$
2. Form the range sketch: $\boldsymbol{Y} = \boldsymbol{A\Omega}$
3. Orthogonalize the columns of the sketch: $\boldsymbol{Y} = \boldsymbol{QR}$

Claim: $\boldsymbol{A} \approx \boldsymbol{QQ}^* \boldsymbol{A}$

Sources: NLA community: Stewart (1970s). GFA: Johnson & Lindenstrauss (1984) et seq. TCS: Boutsidis, Cohen, Deshpande, Drineas, Frieze, Kannan, Mahoney, Nelson, Papadimitriou, Sarlós, Vempala (1998–present). SciComp: Halko, Liberty, Martinsson, Rokhlin, Szlam, Tropp, Tygert, Woolfe (2004–present). See HMT 2011, §2 for more history.

Analysis of the Randomized Range Finder

Theorem 1 (HMT 2011). **Assume**

- *The input matrix* $\boldsymbol{A} \in \mathbb{C}^{m \times n}$
- *Range sketch* $\boldsymbol{Y} = \boldsymbol{A\Omega}$ *where* $\boldsymbol{\Omega} \in \mathbb{C}^{n \times k}$ *is complex standard normal*
- *Form* $\boldsymbol{Y} = \boldsymbol{QR}$ *where* $\boldsymbol{Q}$ *has* k *orthonormal columns*

Then

$$\mathbb{E}_{\boldsymbol{\Omega}} \left\| \boldsymbol{A} - \boldsymbol{QQ}^* \boldsymbol{A} \right\|_{\mathrm{F}} \quad \leq \quad \min_{\varrho < k} \quad \sqrt{\frac{\varrho}{k - \varrho}} \cdot \left\| \boldsymbol{A} - [\![\boldsymbol{A}]\!]_{\varrho} \right\|_{\mathrm{F}}$$

- **Key Fact:** Approximation exploits spectral decay
- Probability of a much larger error is negligible
- Related results hold for the spectral norm

Source: Halko et al. 2011, §10.2; Gu 2015; TYUC 2016–2019.

Overview of SketchySVD

A Tripartite Sketch

- Let $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ be an input matrix (presented in turnstile model)
- Fix sketch size parameter k with $r \leq k \ll \min\{m, n\}$
- Draw independent random linear maps:

$$\boldsymbol{\Upsilon} \in \mathbb{C}^{k \times m} \quad \text{and} \quad \boldsymbol{\Omega} \in \mathbb{C}^{n \times k}$$

$$\boldsymbol{\Phi} \in \mathbb{C}^{2k \times m} \quad \text{and} \quad \boldsymbol{\Psi} \in \mathbb{C}^{n \times 2k}$$

- **Co-range and range sketches:**

$$\boldsymbol{X} = \boldsymbol{\Upsilon} \boldsymbol{A} \in \mathbb{C}^{k \times n} \quad \text{and} \quad \boldsymbol{Y} = \boldsymbol{A} \boldsymbol{\Omega} \in \mathbb{C}^{m \times k}$$

- **Core sketch:**

$$\boldsymbol{Z} = \boldsymbol{\Phi} \boldsymbol{A} \boldsymbol{\Psi} \in \mathbb{C}^{2k \times 2k}$$

Sources: Vempala et al. 1998–2000; Drineas et al. 2004–2006; Martinsson et al. 2004–2012; Sarlós 2006; Woolfe et al. 2008; Clarkson & Woodruff 2009, 2011; Boutsidis et al. 2011, 2016; HMT 2011; Mahoney 2012; Nelson et al. 2012–2015; Woodruff 2014; Gu 2015; Cohen et al. 2015; Upadhyay 2016; TYUC 2016–2019....

The SketchySVD Procedure

1. Use range sketches $\boldsymbol{X}, \boldsymbol{Y}$ to find orthonormal $\boldsymbol{Q} \in \mathbb{C}^{m \times k}$ and $\boldsymbol{P} \in \mathbb{C}^{n \times k}$ where

$$\boldsymbol{A} \quad \approx \quad \boldsymbol{Q}\boldsymbol{Q}^* \boldsymbol{A}\boldsymbol{P}\boldsymbol{P}^*$$

2. Use core sketch $\boldsymbol{Z} \in \mathbb{C}^{2k \times 2k}$ to find core approximation $\boldsymbol{C} \in \mathbb{C}^{k \times k}$ such that

$$\boldsymbol{C} \quad \approx \quad \boldsymbol{Q}^* \boldsymbol{A}\boldsymbol{P}$$

3. For $r \le k$, apply classical or randomized TSVD algorithm to form

$$[\![\boldsymbol{C}]\!]_r \quad = \quad \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$$

4. Obtain approximate r-truncated SVD $\hat{\boldsymbol{A}}_r$ in factored form:

$$\hat{\boldsymbol{A}}_r \quad := \quad \boldsymbol{Q}[\![\boldsymbol{C}]\!]_r \boldsymbol{P}^* \quad = \quad (\boldsymbol{Q}\boldsymbol{U})\boldsymbol{\Sigma}(\boldsymbol{P}\boldsymbol{V})^*$$

Sources: Vempala et al. 1998–2000; Drineas et al. 2004–2006; Martinsson et al. 2004–2012; Sarlós 2006; Woolfe et al. 2008; Clarkson & Woodruff 2009, 2011; Boutsidis et al. 2011, 2016; HMT 2011; Mahoney 2012; Nelson et al. 2012–2015; Woodruff 2014; Gu 2015; Cohen et al. 2015; Upadhyay 2016; TYUC 2016–2019....

Pseudocode for SketchySVD

Input: Sketch size parameters; input matrix $\boldsymbol{A} \in \mathbb{C}^{m \times n}$ as a turnstile stream
Output: Rank-r approximation $\hat{\boldsymbol{A}}_r = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$

function INITIALIZE(m, n, k) ▷ Set up the sketch
    Draw random linear maps $\boldsymbol{\Upsilon}, \boldsymbol{\Omega}, \boldsymbol{\Phi}, \boldsymbol{\Psi}$
    $\boldsymbol{X} \leftarrow \mathbf{0}$ and $\boldsymbol{Y} \leftarrow \mathbf{0}$ and $\boldsymbol{Z} \leftarrow \mathbf{0}$
function LINEARUPDATE$(\boldsymbol{H})$ ▷ Process $\boldsymbol{A} \leftarrow \boldsymbol{A} + \boldsymbol{H}$
    $\boldsymbol{X} \leftarrow \boldsymbol{X} + \boldsymbol{\Upsilon}\boldsymbol{H}$
    $\boldsymbol{Y} \leftarrow \boldsymbol{Y} + \boldsymbol{H}\boldsymbol{\Omega}$
    $\boldsymbol{Z} \leftarrow \boldsymbol{Z} + \boldsymbol{\Phi}\boldsymbol{H}\boldsymbol{\Psi}$
function SKETCHYSVD(r) ▷ Compute r-truncated SVD
    $\boldsymbol{Q} \leftarrow \texttt{economy\_qr}(\boldsymbol{Y})$ ▷ Basis for range
    $\boldsymbol{P} \leftarrow \texttt{economy\_qr}(\boldsymbol{X}^*)$ ▷ Basis for co-range
    $\boldsymbol{C} \leftarrow ((\boldsymbol{\Phi}\boldsymbol{Q}) \backslash \boldsymbol{Z}) / (\boldsymbol{P}^*\boldsymbol{\Psi})$ ▷ Core matrix
    $(\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}) \leftarrow \texttt{svd}(\boldsymbol{C}; r)$ ▷ Truncate dense SVD of core
    $\boldsymbol{U} \leftarrow \boldsymbol{Q}\boldsymbol{U}$ and $\boldsymbol{V} \leftarrow \boldsymbol{P}\boldsymbol{V}$ ▷ Consolidate unitary factors
    return $(\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V})$

Analysis of SKETCHYSVD

Theorem 2 (TYUC 2018). **Assume**

- *The input matrix* $\boldsymbol{A} \in \mathbb{C}^{m \times n}$
- *The dimension reduction maps are independent complex standard normal*

Then *SKETCHYSVD computes a rank-*r *approximation* $\hat{\mathbf{A}}_r$ *for which*

$$\mathbb{E}\left\|\boldsymbol{A}-\hat{\boldsymbol{A}}_r\right\|_{\mathrm{F}} \quad \leq \quad \left\|\boldsymbol{A}-[\![\boldsymbol{A}]\!]_r\right\|_{\mathrm{F}} \quad + \quad \min_{\varrho<k} \quad \sqrt{8 \cdot \frac{k+\varrho}{k-\varrho}} \cdot \left\|\boldsymbol{A}-[\![\boldsymbol{A}]\!]_\varrho\right\|_{\mathrm{F}}$$

- **Key Fact:** Approximation exploits spectral decay
- Probability of a much larger error is negligible
- Related results hold for the spectral norm

Source: HMT 2011; Gu 2015; TYUC 2016–2019.

Resource Usage with Sparse Maps

Storage:

- Random linear maps: $O(m+n)$
- Sketches: $O(k(m+n))$

Arithmetic:

- Linear update: Depends on structure of update (cheap!)
- SKETCHYSVD: $O(k^2(m+n))$
  - Computation of range and co-range: $O(k^2(m+n))$
  - Computation of core: $O(k(m+n)+k^3)$
  - Truncated SVD of core: $O(k^3)$
  - Consolidation: $O(k^2(m+n))$

Communication:

- One pass over data

Sources: Charikar et al. 2002; Cormode & Muthukrishnan 2005; Sarlós 2006; Woolfe et al. 2008; Clarkson & Woodruff 2009, 2011; HMT 2011; Nelson et al. 2012–2015; Meng & Mahoney 2013; Cohen 2015; TYUC 2016–2019....

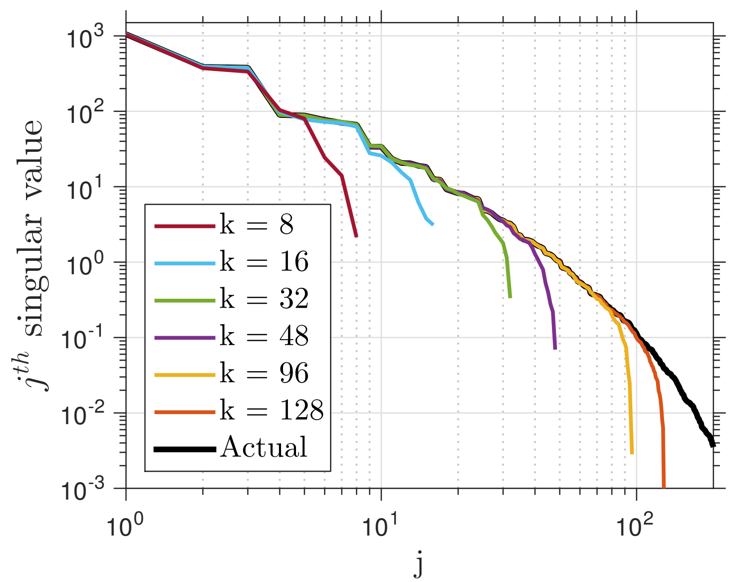
Empirical Performance

Important Things I'm Not Going to Show You

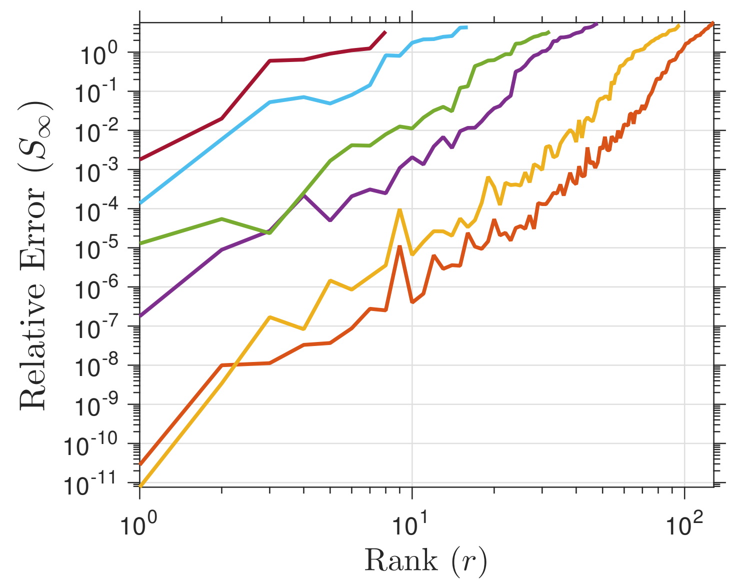
- SKETCHYSVD is insensitive to the choice of random linear map
- Theory gives parameter choices that are nearly optimal in practice
- SKETCHYSVD beats earlier techniques for synthetic and real data
- Methodology for estimating errors and selecting the truncation rank
- Sampling distribution of approximation error and error estimator
- Other structured approximations via re-factorization or matrix nearness
- Transformation of the data before sketching
- Extension to low-rank Tucker approximation of a tensor

Sources: HMT 2011; TYUC 2016–2019; SGLTU 2018–2019.

Why Truncate?



(A) Spectrum of Approximation $\sigma_r(\hat{\boldsymbol{A}}_k)$



(B) Error in Rank-r Truncation $\hat{\boldsymbol{A}}_r$

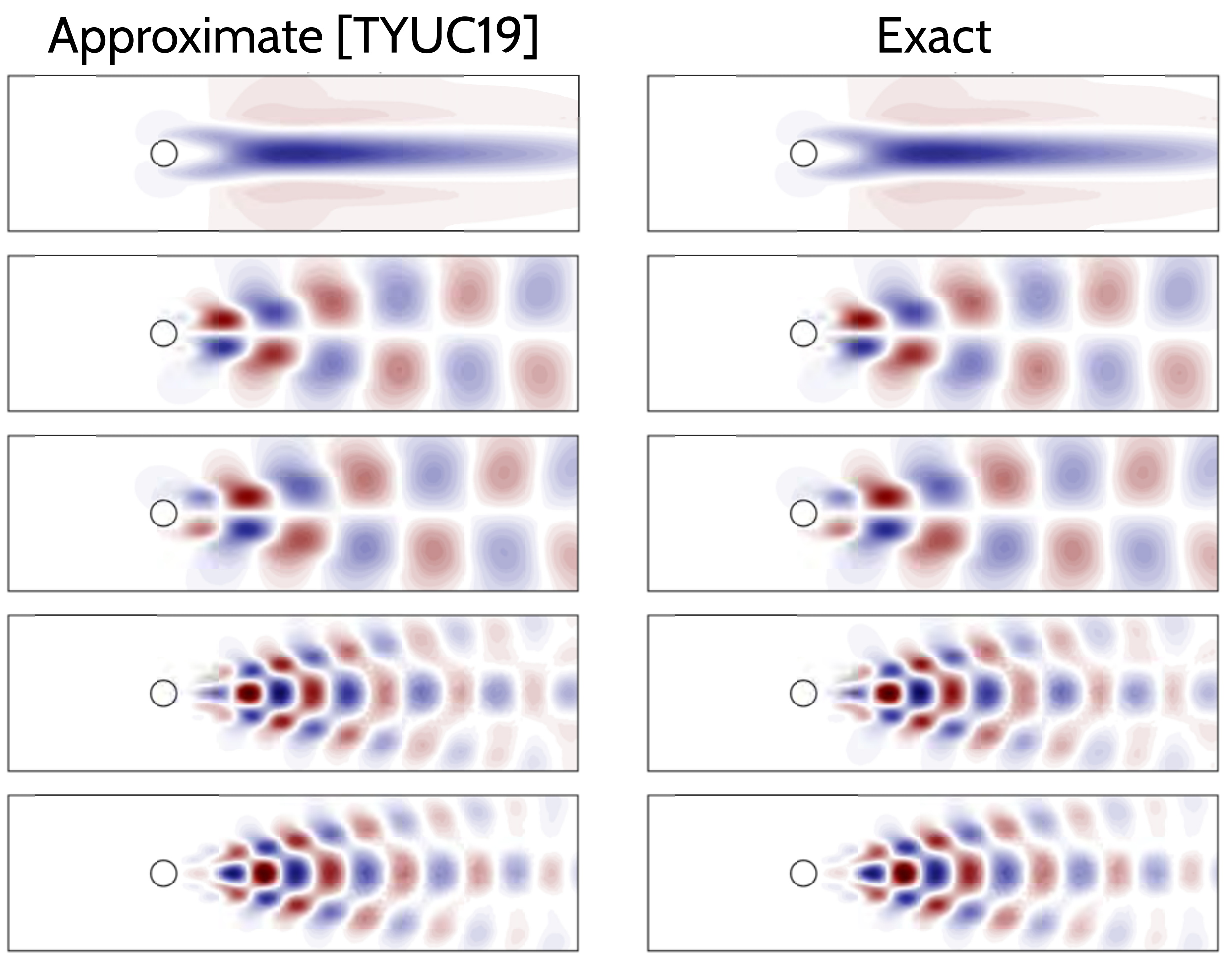
$$\mathrm{relerr}(\hat{\boldsymbol{A}}_r) = \frac{\|\boldsymbol{A} - \hat{\boldsymbol{A}}_r\|}{\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|} - 1$$

Comments: `StreamVel` Data: $m = 10,738$; $n = 5,001$; 430 MB. Algorithm: Sparse maps; $s = 2k + 1$.

Reconstruction of von Kármán Street

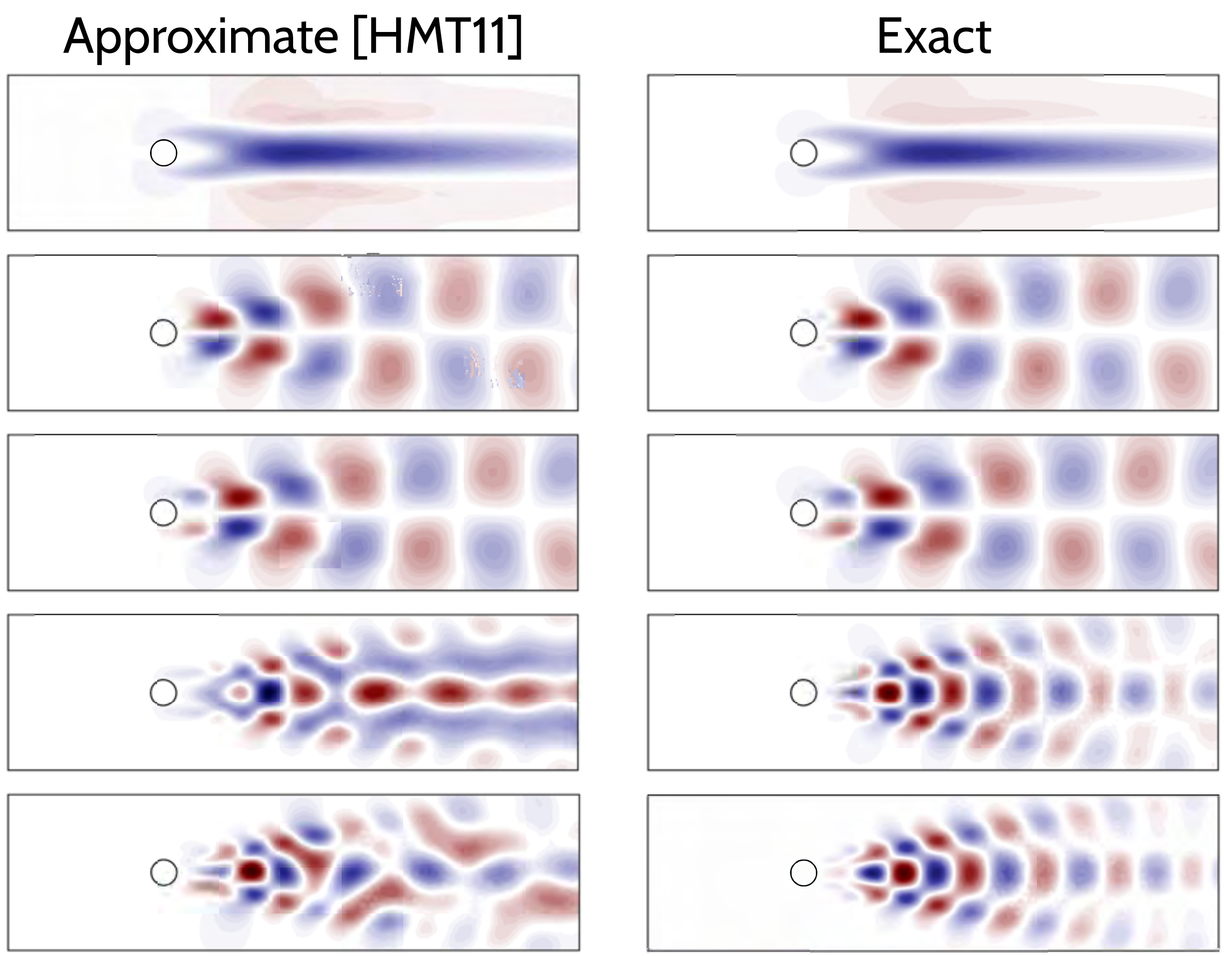
Comments: Data: $m = 10{,}738$; $n = 5{,}001$; 430 MB. Algorithm: Sparse maps; rank $r = 5$; storage $T = 48(m + n)$. Compression: $71\times$.

Left Singular Vectors of von Kármán Street



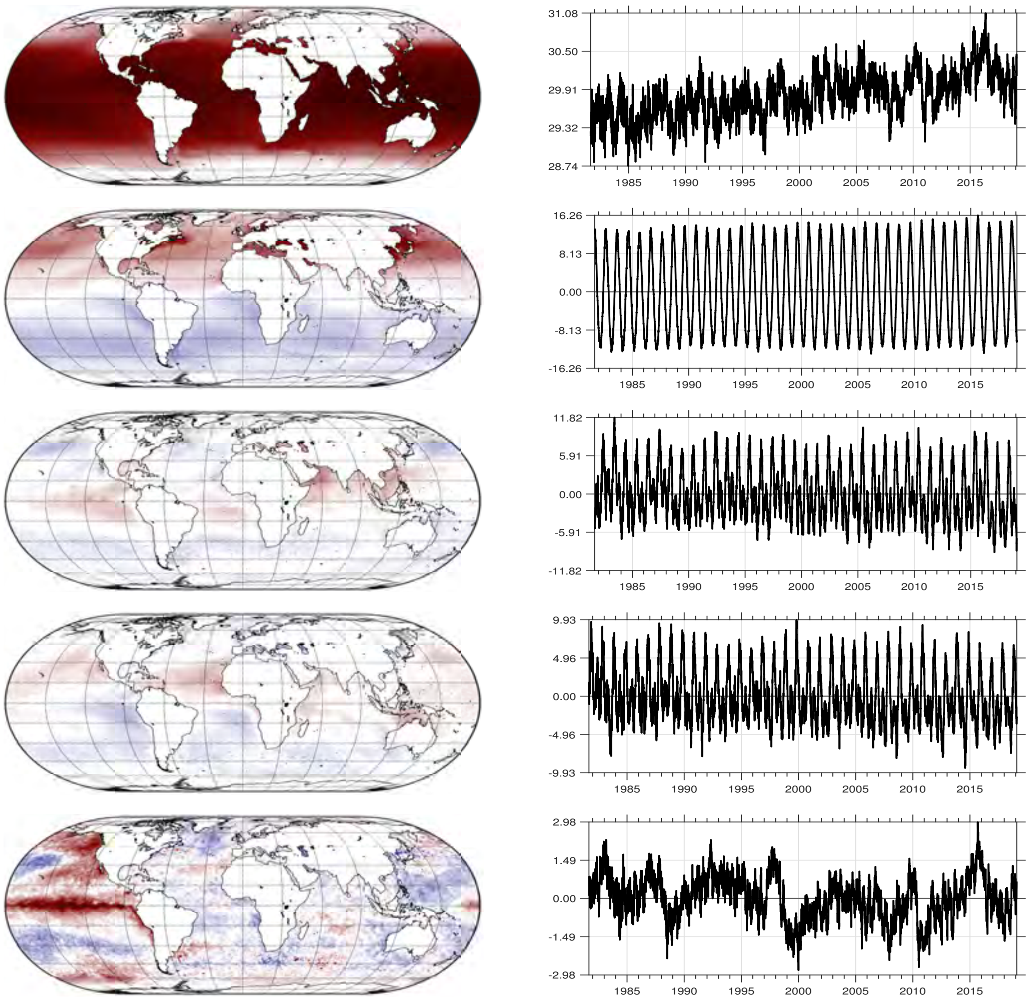
Comments: Data: $m = 10,738$; $n = 5,001$; 430 MB. Algorithm: Sparse maps; rank $r = 5$; storage $T = 48(m+n)$. Compression: $71\times$.

Left Singular Vectors of von Kármán Street



Comments: Data: $m = 10,738$; $n = 5,001$; 430 MB. Algorithm: Sparse maps; rank $r = 5$; storage $T = 48(m + n)$. Compression: $71\times$.

Singular Vectors of Sea Surface Temperature Data



Spatiotemporal Avg.

Seasonal

(Intra-)Seasonal

(Intra-)Seasonal

El Niño / La Niña

Comments: Data: $m = 691,150$; $n = 13,670$; 75 GB. Algorithm: Sparse maps; $k = 48$; $s = 839$. Compression ratio: $222\times$.

To learn more...

E-mail: `jtropp@cms.caltech.edu`

Web: `http://users.cms.caltech.edu/~jtropp`

Papers:

- Halko, Martinsson, & Tropp, "Finding structure with randomness: Probabilistic algorithms for computing approximate matrix decompositions," *SIREV*, 2011.
- Tropp, Yurtsever, Udell, & Cevher, "Sketchy decisions: Low-rank matrix optimization with optimal storage," AISTATS 2017.
- Tropp, Yurtsever, Udell, & Cevher, "Practical sketching algorithms for low-rank matrix approximation," *SIMAX*, 2017.
- Tropp, Yurtsever, Udell, & Cevher, "Fixed-rank approximation of a positive-semidefinite matrix from streaming data," *NeurIPS*, 2017.
- Tropp, Yurtsever, Udell, & Cevher, "Streaming low-rank matrix approximation with an application to scientific simulation," arXiv cs.NA 1902.08651. To appear, *SISC*.
- Sun, Guo, Tropp, & Udell, "Tensor random projections for low-memory dimension reduction," *NeurIPS Relational Databases Workshop*, 2018.
- Sun, Guo, Luo, Tropp, & Udell, "Low-rank Tucker approximation of a tensor from streaming data," arXiv cs.NA 1904.10951
- Cevher, Tropp, & Yurtsever, "Scalable semidefinite programming." Coming soon!

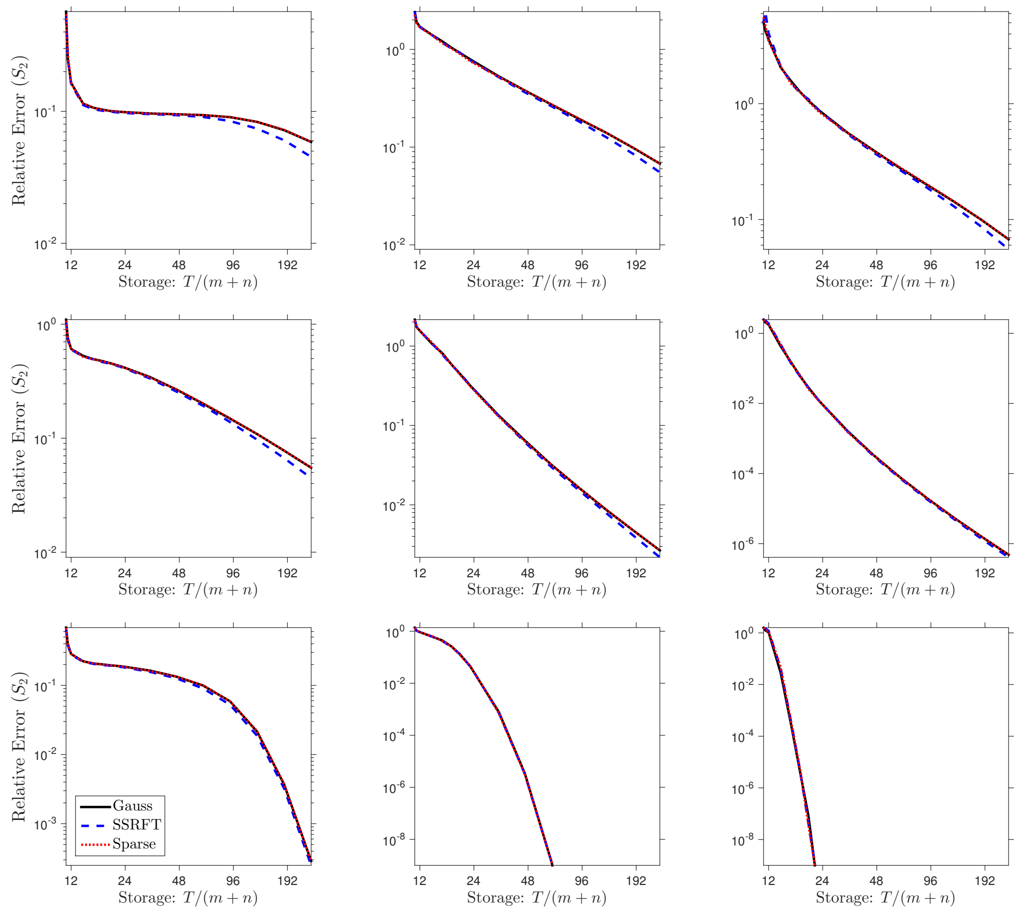
Supplementary Materials

Insensitivity to Dimension Reduction Map

LowRankHiNoise
LowRankMedNoise
LowRankLowNoise

PolyDecaySlow
PolyDecayMed
PolyDecayFast

ExpDecaySlow
ExpDecayMed
ExpDecayFast



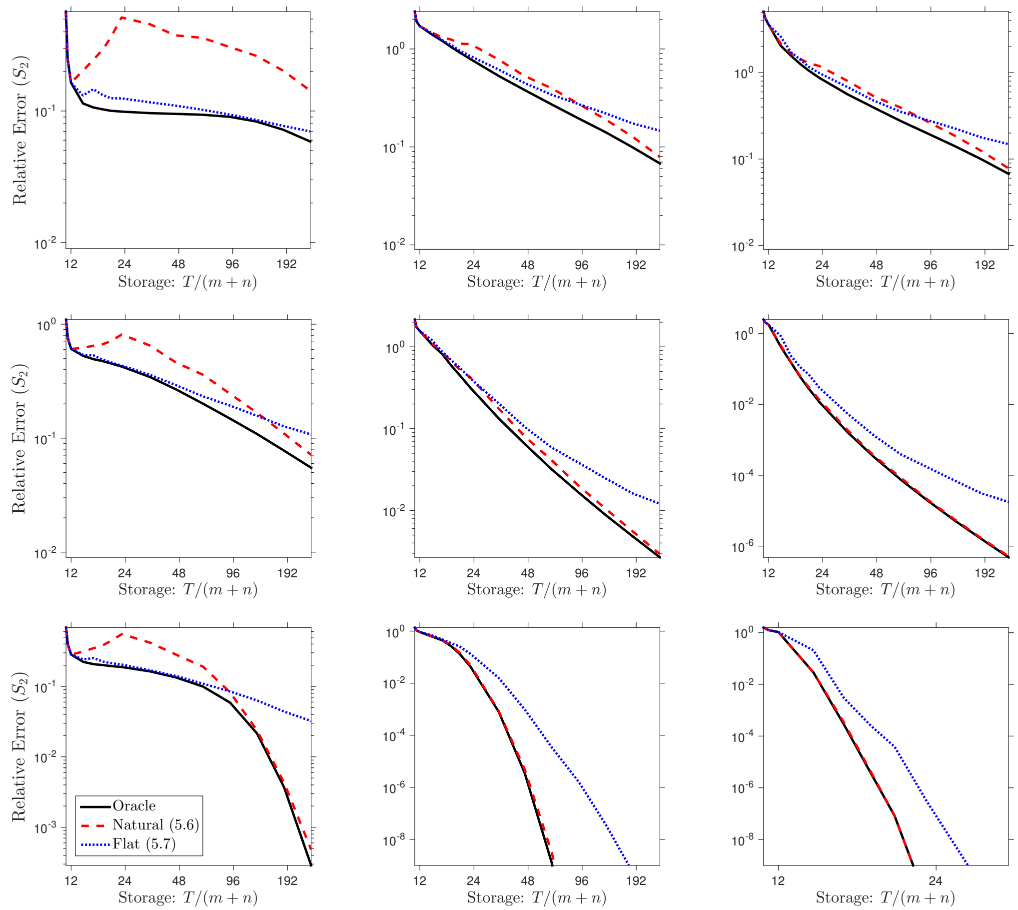
Comments: Effective rank $R = 10$, approximation rank $r = 10$, Schatten 2-norm.

Performance with Theoretical Parameter Choices

LowRankHiNoise
LowRankMedNoise
LowRankLowNoise

PolyDecaySlow
PolyDecayMed
PolyDecayFast

ExpDecaySlow
ExpDecayMed
ExpDecayFast



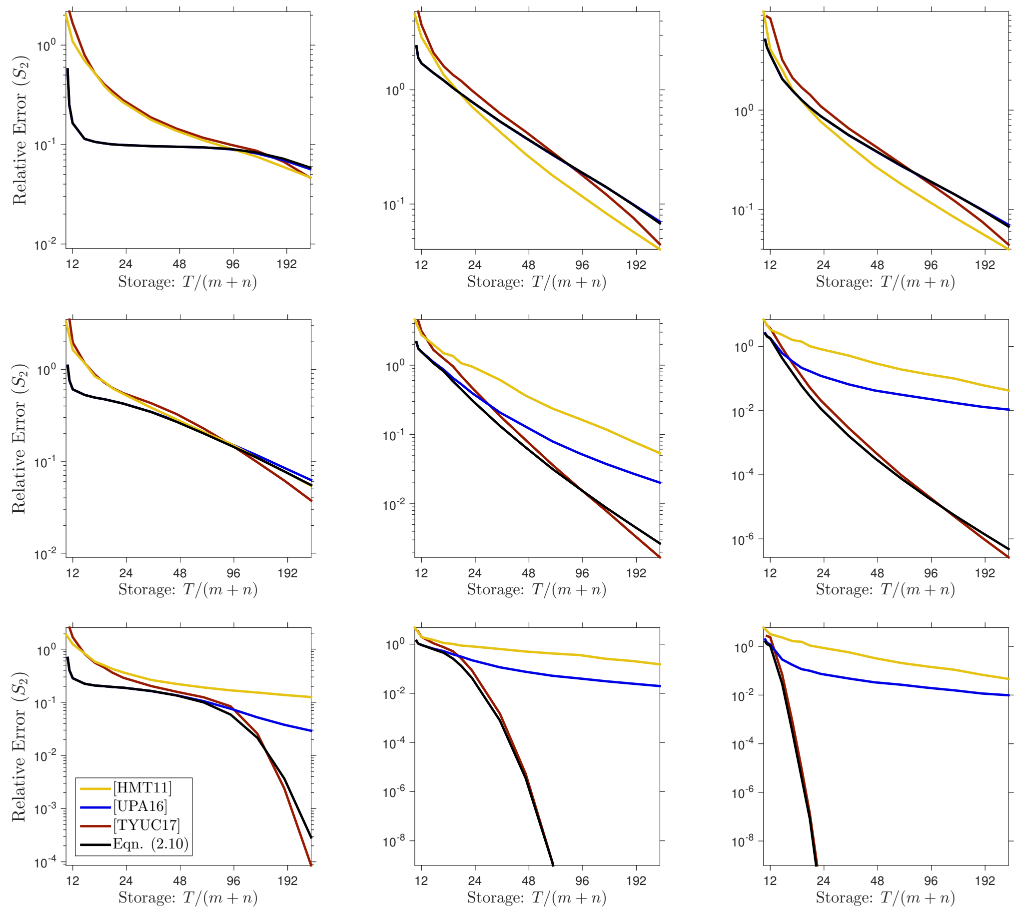
Comments: Gaussian maps, effective rank $R = 10$, approximation rank $r = 10$, Schatten 2-norm.

Method Comparison: Synthetic Data

LowRankHiNoise
LowRankMedNoise
LowRankLowNoise

PolyDecaySlow
PolyDecayMed
PolyDecayFast

ExpDecaySlow
ExpDecayMed
ExpDecayFast

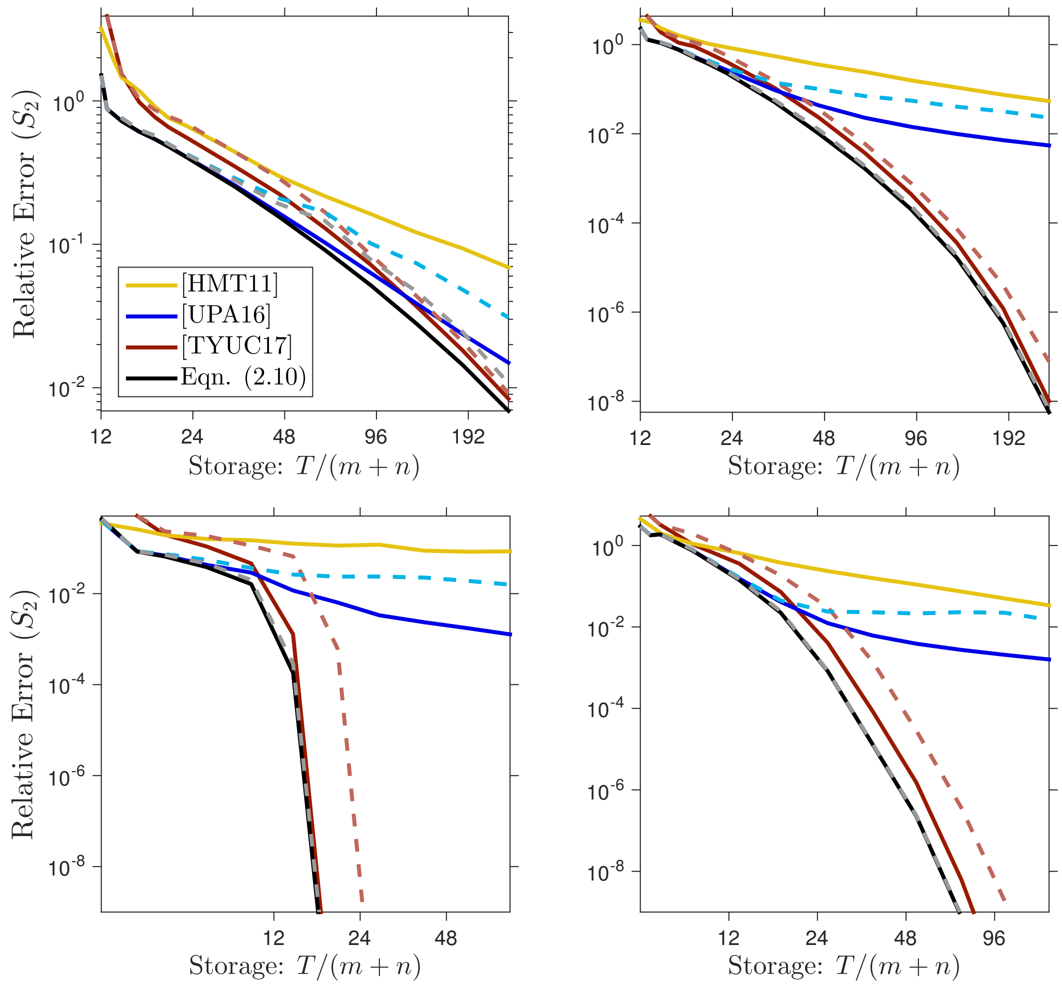


Comments: Gaussian maps, effective rank $R = 10$, oracle parameters, approximation rank $r = 10$, Schatten 2-norm.)

Method Comparison: Real Data

MinTemp
$m = 19{,}264$
$n = 7{,}305$
$r = 10$

StreamVel
$m = 10{,}738$
$n = 5{,}001$
$r = 10$



MaxCut
$n = 2{,}000$
$r = 1$

PhaseRetrieval
$n = 25{,}921$
$r = 5$

Comments: Sparse maps, Schatten 2-norm. Solid lines are errors with oracle parameters; dashed lines are *a priori* parameter choices.

A Posteriori Error Estimation

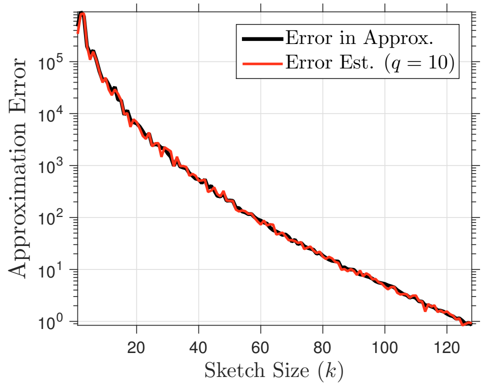
- Fix a sketch size parameter q
- Draw a random Gaussian dimension reduction map $\mathbf{\Theta} \in \mathbb{C}^{m \times q}$
- Maintain an error sketch $\boldsymbol{S} = \mathbf{\Theta}\boldsymbol{A}$
- Given an approximation $\hat{\boldsymbol{A}}$, compute the error estimator

$$\operatorname{err}_2^2(\hat{\boldsymbol{A}}) = \left\| \boldsymbol{S} - \mathbf{\Theta}\hat{\boldsymbol{A}} \right\|_{\mathrm{F}}^2$$

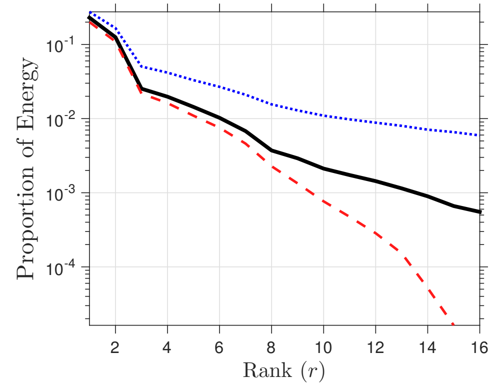
- The error estimator is unbiased and concentrates sharply
- We can also compute an empirical upper bound on the scree curve as

$$\overline{\operatorname{scree}}(r) = \left[\frac{\tau_{r+1}(\hat{\boldsymbol{A}}) + \operatorname{err}_2(\hat{\boldsymbol{A}})}{\operatorname{err}_2(\mathbf{0})} \right]^2.$$

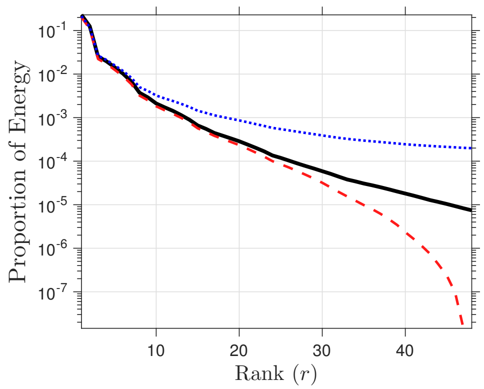
Error Estimates and Empirical Scree Curves



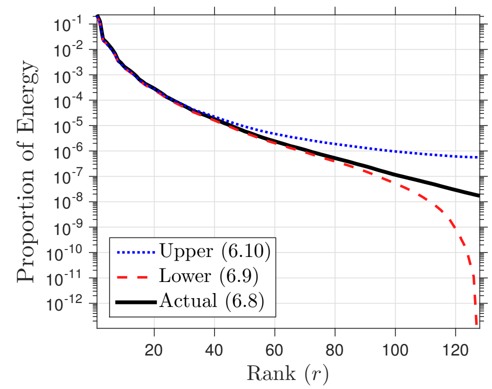
(A) Error Estimates for $\hat{\boldsymbol{A}}$



(B) Scree Plot ($k = 16$)



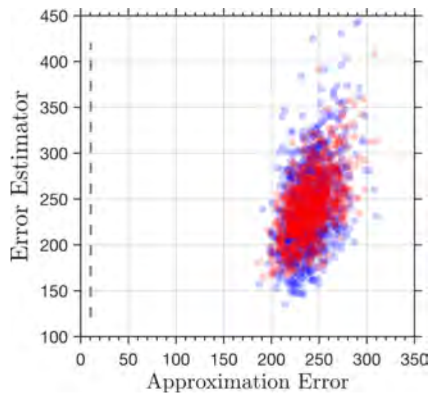
(C) Scree Plot ($k = 48$)



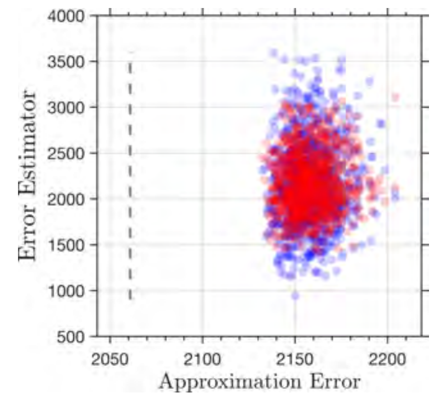
(D) Scree Plot ($k = 128$)

Comments: `StreamVel`, sparse maps, $s = 2k + 1$, $q = 10$.

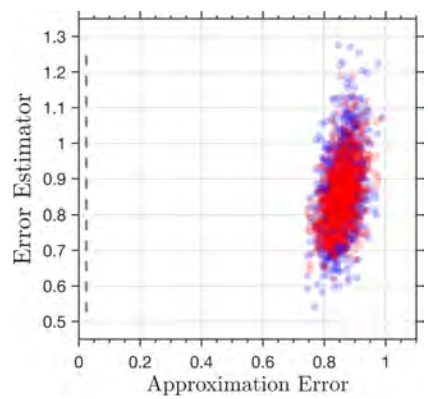
Sampling Distribution of Error and Estimator



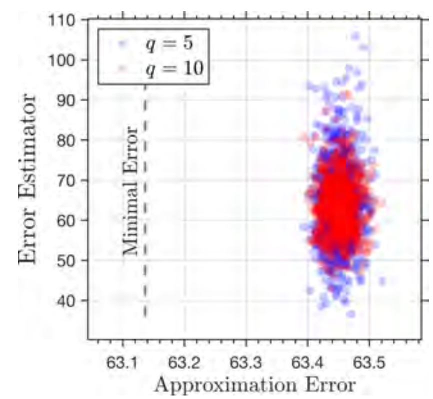
(A) Rank-k Approximation ($k = 48$)



(B) Rank-r Truncation ($k = 48$, $r = 12$)



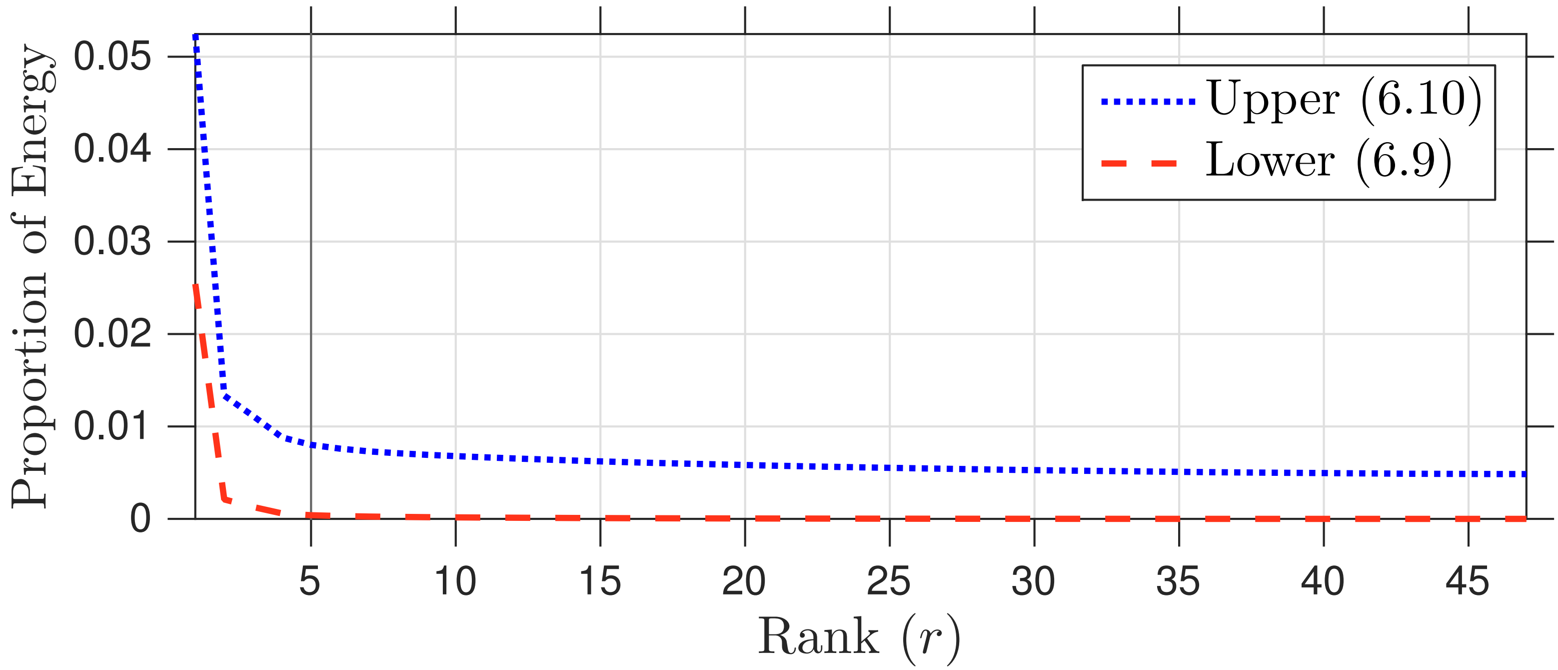
(C) Rank-k Approximation ($k = 128$)



(D) Rank-r Truncation ($k = 128$, $r = 32$)

Comments: `StreamVel`, sparse maps, $s = 2k + 1$.

Scree Plot for Sea Surface Temperature Data



Comments: `SeaSurfaceTemp`, sparse maps, $k = 48$, $s = 839$, $q = 10$