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)
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)

\[ A \approx U_r \Sigma_r V^*_r \]

- \( U, V \) have orthonormal columns and \( \Sigma \) 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:

\[ A = H_1 + H_2 + H_3 + H_4 + \cdots \]

- Huge input matrix \( A \) is presented as a sum of innovations \( H_i \)
- Must discard each innovation \( H_i \) after it is processed

**Goal:** Without storing \( A \) 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

\[ \text{sketch} = \mathcal{L}(A) = \sum_i \mathcal{L}(H_i) \]

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

**Examples:**

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

**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

\[ A \omega_1 \]

\[ A \omega_2 \]
The Randomized Range Finder

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

A. Just multiply random vectors into $A$ and orthogonalize!

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

Claim: $A \approx QQ^* A$

Analysis of the Randomized Range Finder

Theorem 1 (HMT 2011). Assume

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

Then

\[
E_\Omega \| A - QQ^* A \|_F \leq \min_{\varrho < k} \sqrt{\frac{\varrho}{k - \varrho}} \cdot \| A - [A]_\varrho \|_F
\]

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

Overview of SketchySVD
A Tripartite Sketch

- Let $A \in \mathbb{C}^{m \times n}$ 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:

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

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

- Co-range and range sketches:

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

- Core sketch:

  \[ Z = \Phi A\Psi \in \mathbb{C}^{2k \times 2k} \]

The SKETCHYSVD Procedure

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

$$A \approx QQ^* APP^*$$

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

$$C \approx Q^* AP$$

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

$$[C]_r = U \Sigma V^*$$

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

$$\hat{A}_r := Q[C]_r P^* = (QU) \Sigma (PV)^*$$

Pseudocode for SKETCHYSVD

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

1. function `INITIALIZE(m, n, k)`  
   - Draw random linear maps $\Upsilon, \Omega, \Phi, \Psi$  
   - $X \leftarrow 0$ and $Y \leftarrow 0$ and $Z \leftarrow 0$  
   - Set up the sketch

2. function `LINEARUPDATE(H)`  
   - $X \leftarrow X + \Upsilon H$  
   - $Y \leftarrow Y + H\Omega$  
   - $Z \leftarrow Z + \Phi H\Psi$  
   - Process $A \leftarrow A + H$

3. function `SKETCHYSVD(r)`  
   - $Q \leftarrow \text{economy}_\text{qr}(Y)$  
   - $P \leftarrow \text{economy}_\text{qr}(X^*)$  
   - $C \leftarrow ((\Phi Q) \setminus Z) / (P^* \Psi)$  
   - $(U, \Sigma, V) \leftarrow \text{svd}(C; r)$  
   - $(U, \Sigma, V) \leftarrow \text{svd}(C; r)$  
   - Compute $r$-truncated SVD  
   - Basis for range  
   - Basis for co-range  
   - Core matrix

4. $U \leftarrow QU$ and $V \leftarrow PV$  
   - Truncate dense SVD of core  
   - Consolidate unitary factors

5. return $(U, \Sigma, V)$

---

Joel A. Tropp (Caltech), SketchySVD, Geometry and Big Data, IPAM, 2 May 2019
Analysis of SKETCHYSVD

Theorem 2 (TYUC 2018). Assume

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

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

$$\mathbb{E} \left\| A - \hat{A}_r \right\|_F \leq \left\| A - [A]_r \right\|_F + \min_{\rho < k} \sqrt{8 \cdot \frac{k + \rho}{k - \rho}} \cdot \| A - [A]_{\rho} \|_F$$

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

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

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

Why Truncate?

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

\[
\text{relerr}(\hat{A}_r) = \frac{\|A - \hat{A}_r\|}{\|A - [A]_r\|} - 1
\]

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

Joel A. Tropp (Caltech), SketchySVD, Geometry and Big Data, IPAM, 2 May 2019
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

Approximate [TYUC19]                             Exact

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

Approximate [HMT11]  

Exact

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

Joel A. Tropp (Caltech), SketchySVD, Geometry and Big Data, IPAM, 2 May 2019
Singular Vectors of Sea Surface Temperature Data

Spaitotemporal 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:

- 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.

Joel A. Tropp (Caltech), *SketchySVD, Geometry and Big Data*, IPAM, 2 May 2019
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.

Joel A. Tropp (Caltech), SketchySVD, Geometry and Big Data, IPAM, 2 May 2019
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.)

Joel A. Tropp (Caltech), SketchySVD, Geometry and Big Data, IPAM, 2 May 2019
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 \textit{a priori} parameter choices.
A Posteriori Error Estimation

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

$$\text{err}_2^2(\hat{A}) = \| S - \Theta \hat{A} \|_F^2$$

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

$$\text{scree}(r) = \left[ \frac{\tau_{r+1}(\hat{A}) + \text{err}_2(\hat{A})}{\text{err}_2(0)} \right]^2.$$
Error Estimates and Empirical Scree Curves

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

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

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

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

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

Joel A. Tropp (Caltech), SketchySVD, Geometry and Big Data, IPAM, 2 May 2019
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, $s = 2k + 1$. 

Joel A. Tropp (Caltech), SketchySVD, Geometry and Big Data, IPAM, 2 May 2019
Scree Plot for Sea Surface Temperature Data

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