Randomized SVD (rSVD)

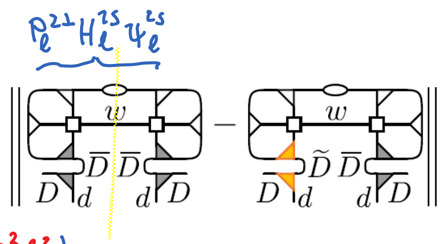
[Halko2011] (highly readable paper!)

rSVD.1

Shrewd selection was proposed to avoid an expensive SVD:

Goal: truncate $\overline{A}_\ell(\overset{\bar{D}}{\blacktriangledown}) \to \widetilde{A}^{\rm tr}_\ell(\overset{\tilde{D}}{\blacktriangledown})$ to minimize $C_1 =$

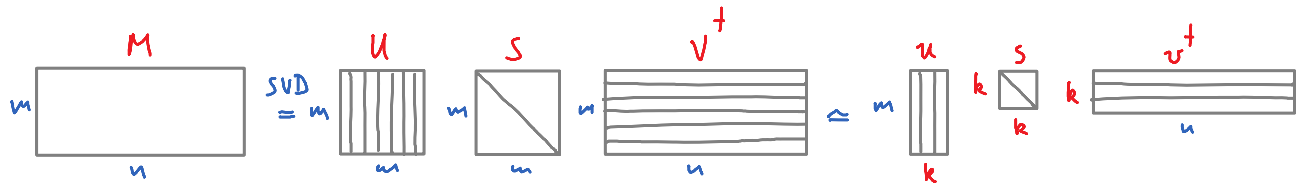
orthogonal complement — truncated complement



Optimal truncation can be achieved via SVD; but that has 2s costs, $\mathcal{O}(D^3 d^3)$

[McCullogh2024] pointed out: a more generic approach to avoid an expensive SVD is a 'randomized SVD' (rSVD).

Consider $m \times n$ matrix M. Cost of full SVD: $\mathcal{O}(m \cdot n \ \min(m,n))$ (my figures assume m < n)



(1)

If we know that we will truncate it to rank $k \ll m, n$, computing full SVD is wasteful!

rSVD offers a way of finding truncated SVD at costs $\mathcal{O}(m\, n \cdot (k+p))$ (2)

target rank — oversampling parameter

Definition: 'range' of a matrix is the vector space spanned by its column vectors.

Matrix-vector multiplication yields 'linear combination of column vectors' = 'vector in range of matrix'

$\vec{y} = M\vec{x} = \vec{c}_j\, x^j \in \text{range}(M)$ $\qquad y^i = M^i{}_j\, x^j = (c_j)^i\, x^j$ (3)

column j of M $\qquad$ element i of column j of M

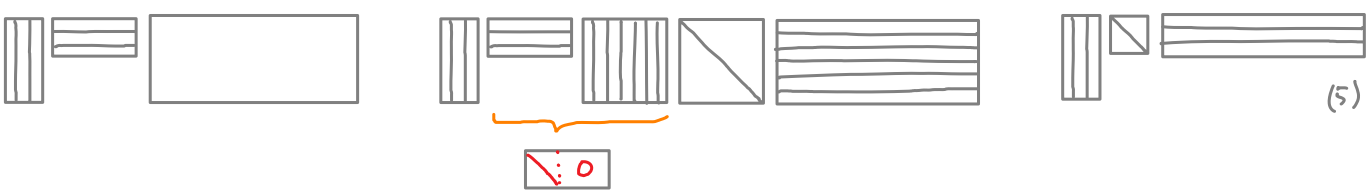
For a truncated SVD, the range of u is the 'most relevant' k -dimensional subspace of range of M

$\vec{y} = u s v^\dagger \vec{x} \Rightarrow \vec{c}_j\, (s v^\dagger \vec{x})^j \in \text{range}(u)$ (4)

column j of u

The 'truncated' version of M can be found by projection onto the range of u :

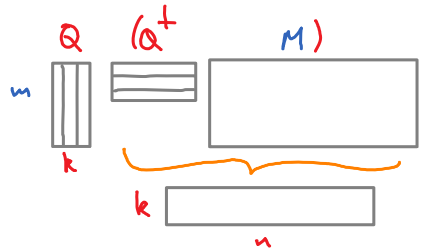
$u u^\dagger \cdot M = u u^\dagger\, \underbrace{U}\cdot S\, V^\dagger = u\, s\, v^\dagger$



(5)

Suppose Q is a good guess for u. Then SVD that truncates M can be found cheaply via full SVD of $Q^\dagger M$:

Suppose Q is a good guess for u. Then SVD that truncates M can be found cheaply via full SVD of $Q^\dagger M$.



(6)

Cost of SVD: $\mathcal{O}(k^2 n)$ (7)

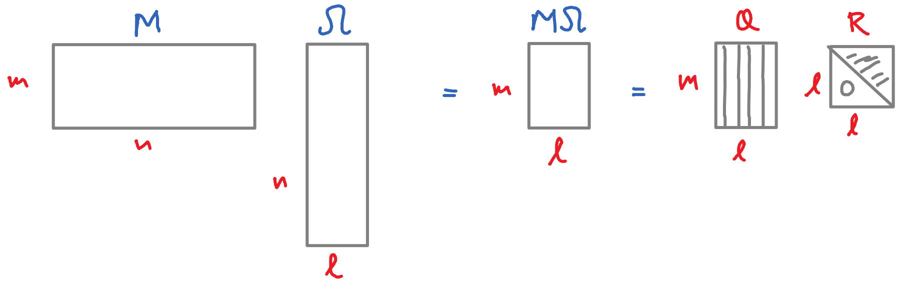
Key idea of randomized SVD: find good guess for u by sampling range of M using random input vectors $\vec{x}$

'Range finder algorithm':

(i) Construct random $n \times \ell$ 'test matrix' Ω, with $\ell = \underbrace{k}_{\text{target rank}} + \underbrace{p}_{\text{oversampling parameter}} < m, n$ (8)

(ii) Compute $M\Omega$ Cost: $\mathcal{O}(m \cdot n \cdot \ell)$, $\dim(M\Omega) \simeq \ell$ (9)

(iii) Do thin QR-decomposition $M\Omega = QR$ (10)



(11)

Since columns of Ω are random vectors, the columns of $M\Omega$ are very likely linearly independent.
Then, Q has ℓ columns. They 'explore' (try to 'find') the range of M, thus serve as good guess for u.

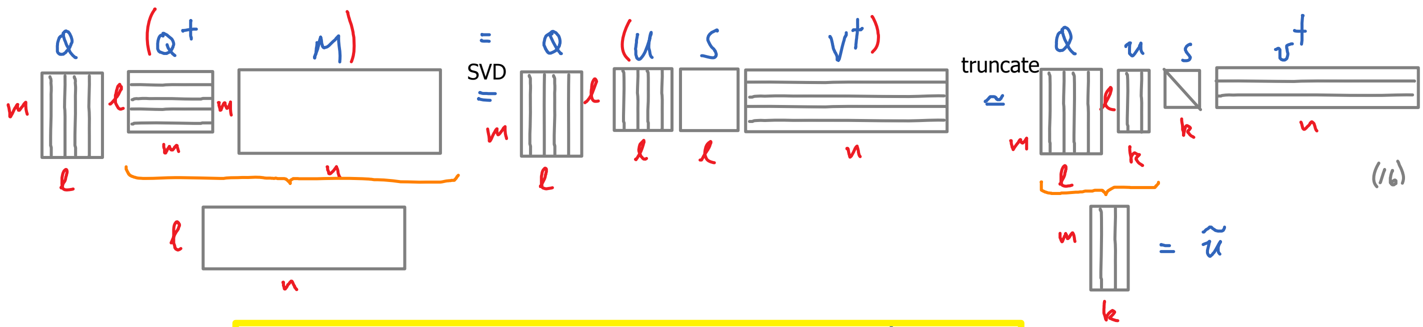
'Subsequent factorization': (compare (6)):

(iv) Compute $Q^\dagger M$ (12)

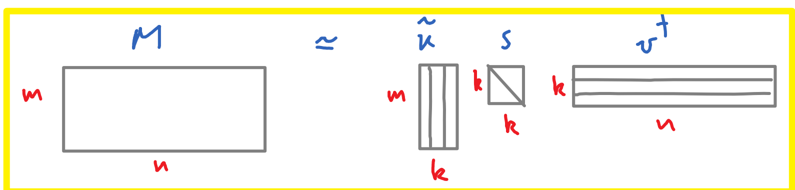
(v) Perform full SVD on $Q^\dagger M = USV^\dagger$ and truncate, $\simeq u s v^\dagger$, from $\ell = k + p$ to k singular values. (13)

(vi) Construct $\tilde{u} = Q u$ (14)

Final result: rSVD of M is given by $M \simeq \tilde{u} s v^\dagger$ (15)



(16)



(17)

Remarks:

1. Total cost: $\mathcal{O}(m \cdot n \cdot \ell)$ Sophisticated implementation can yield lower costs, see [Halko2011].

2. Accuracy:

For full SVD + truncation to rank k : $\| M - u u^\dagger M \| = s_{k+1}$

$\| \ \|$ = ℓ_2 operator norm = largest singular value

first discarded singular value of M

For rSVD with $\ell = k + p$: $\mathbb{E} \| M - Q Q^\dagger M \| \le \left[1 + \frac{4\sqrt{k+p}}{p-1} \sqrt{\min\{m,n\}} \right] s_{k+1}$

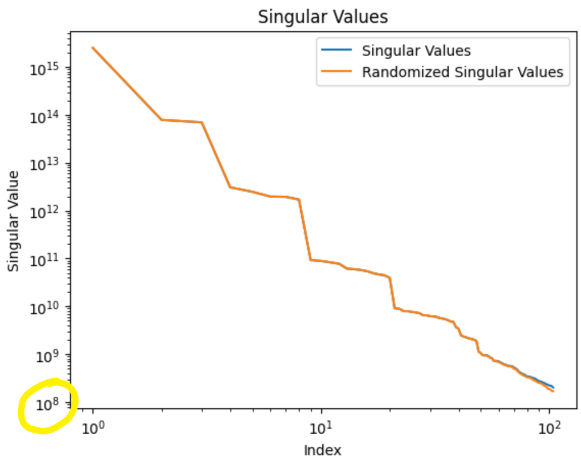
$\mathbb{E}$ = expectation value w.r.t. sampling over random test matrices

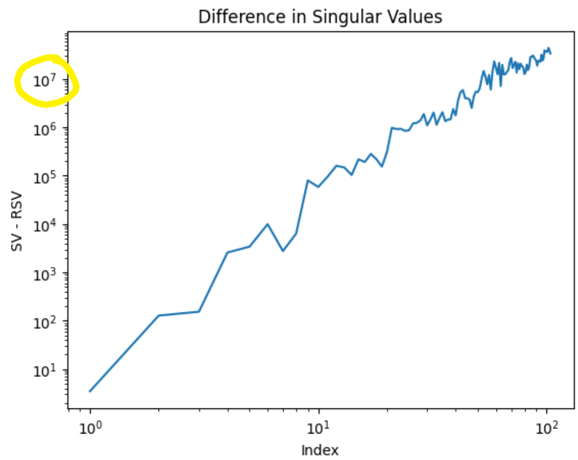
3. Error probability decreases rapidly when increasing oversampling parameter p :

$$P\left(\| M - Q Q^\dagger A \| > \left[1 + 9\sqrt{k+p} \cdot \sqrt{\min\{m,n\}} \right] s_{k+1} \right) < 6 \cdot p^{-p}$$

In practice, $p = 5$ suffices ($1 - 3 \cdot 5^{-5} = 0.99904$)

4. Example: M = random matrix with $m = n = 200$, rSVD with $k = 100$, $p = 5$





5. rSVD is advisable in variational contexts, i.e. during sweeps, where small errors made at a given iteration can be compensated by doing additional iterations.

6. Try using rSVD yourself in your MPS computations! Write a rSVD routine, replace SVD by rSVD.