Effectiveness of the Randomized SVD

Brady Metherall

25 November 2019

1 Introduction

The singular value decomposition (SVD) factors a matrix into a unitary matrix, U, a non-negative diagonal matrix, Σ, and another unitary matrix, V^T. The elements of Σ are ordered in decreasing order and are called the singular values. In Section 2 we investigate how a matrix can be approximated—and compressed—by the truncated SVD, then in Section 3 we explore a much faster method of approximating a matrix and show that this produces a similar error as truncated SVD. Finally, in Section 4 we give our closing remarks.

2 Truncated SVD

Often times the SVD sheds light on the underlying properties of its corresponding matrix, in a similar way as principle component analysis (PCA). For example, if several of the singular values are very small we may omit these to compress the matrix. Such a process is called the truncated SVD where only the r largest singular values are kept and the rest are discarded along with the associated columns of U and V. This provides us with a rank-r approximation to the original matrix. If r is less than the reduced mass of m and n, the truncated SVD will provide a method of compression as the three matrices of SVD will require less memory than the full matrix. In fact, truncated SVD yields the best rank-r approximation, this can be proved by the following two theorems.

Theorem 1. *If AB^T and B^TA are both symmetric matrices, then and only then can two orthogonal matrices U and V be found such that $\Sigma_A = U^TAV$, and $\Sigma_B = U^TBV$ are both diagonal matrices.*

Proof. The proof follows from the spectral theorem [1]. □

Theorem 2 (Eckart–Young Theorem)**.** *The best (in the Frobenius norm) rank-r approximation to a matrix, A, is obtained by the truncated SVD, A_r.*

Proof. The following proof has been adapted from [2]. The best approximation, M, can be found by

$$M = \underset{X \in \bowtie_r}{\arg\min} \|A - X\|_F^2, \tag{1}$$

where $\bowtie_r$ is the set of all rank-r $m \times n$ matrices. The minimum error is then

$$\begin{aligned} \|A - M\|_F^2 &= \langle A, A\rangle_F - 2\langle A, M\rangle_F + \langle M, M\rangle_F, \\ &= \langle A, A\rangle_F - 2\big\langle A, U\Sigma_M V^T\big\rangle_F \\ &\qquad + \langle \Sigma_M, \Sigma_M\rangle_F. \end{aligned} \tag{2}$$

At the minimum, the change in $\|A - X\|_F^2$ is zero for some change in X. This change in X can be encapsulated as $U \mapsto sU$ where s is infinitesimal and antisymmetric to maintain orthogonality. Thus, at the minimum

$$0 = \langle A, sM\rangle_F = \big\langle AM^T, s\big\rangle_F. \tag{3}$$

Therefore, it is the case that AM^T is symmetric. By following a similar procedure, we find that M^TA must be symmetric as well.

By Theorem 1, A and M exhibit the same U and V in their SVDs. Now the error can be simplified to

$$\|A - M\|_F^2 = \|\Sigma_A - \Sigma_M\|_F^2, \tag{4}$$

$$= \sum_{i=1}^{n} (\sigma_i(A) - \sigma_i(M))^2, \tag{5}$$

$$= \sum_{i=r+1}^{n} \sigma_i(A)^2. \tag{6}$$

This minimum is indeed achieved by the truncated SVD, since $\sigma_i(A_r) = \sigma_i(A)H(r - i)$. □

Additionally, the analysis in Section 3 will be dependent on:

Remark 1. *Theorem 2 was extended to all unitarily invariant norms in 1960 by Mirsky [3].*

Although truncated SVD can provide a more compressed version of a matrix, it can be rather expensive to compute as the full SVD is required. To find a similar decomposition in faster time we turn our attention to the randomized SVD.

Algorithm 1 Randomized SVD.

1: Input: Matrix A of size $m \times n$, Int r, Int l.
2: Output: Matrix of size $m \times n$ of rank-r.
3: $\quad \Omega \leftarrow \mathcal{N}(0,1)^{n\times(r+l)}$
4: $\quad Q,\ \_ \leftarrow qr(A * \Omega)$
5: $\quad U,\ \Sigma,\ V \leftarrow svd(Q^T * A)$
6: **return** $(Q * U)[:, 1{:}r] * \Sigma[1{:}r, 1{:}r] * V[1{:}r, :]$

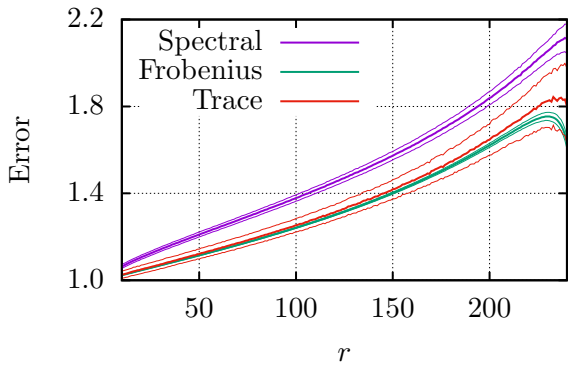
3 Randomized SVD

Randomized SVD attempts to obtain an approximation to the truncated SVD in a fraction of the time at the expensive of the accuracy. We shall investigate the expected value of the relative error for a variety of matrices. If the relative error is $\mathcal{O}(1)$ then the randomized SVD would be much more effective than truncated SVD because of the reduced computation time.

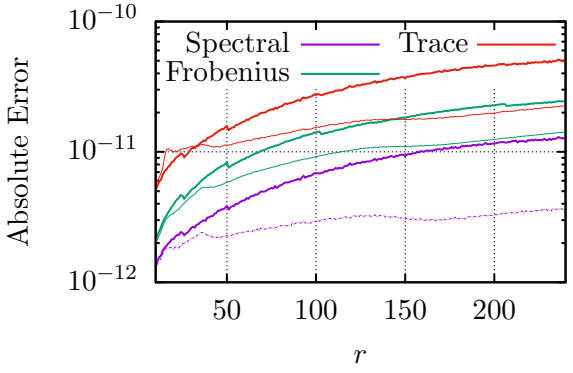
The randomized SVD is found using the method outlined in Algorithm 1. Effectively, our matrix, $A \in \mathbb{R}^{m\times n}$, is first multiplied by a random matrix of rank-$(r+l)$, where r is the rank we wish to approximate, and l is the buffer size. This extracts $r+l$ random directions of A, we then take the QR decomposition of this product to orthonormalize the projections. Finally, we compute the thin SVD on this much smaller matrix.

To have a better understanding of the expected speed up of randomized over truncated SVD we proceed by finding the computational complexity. The complexity of Algorithm 1 can be broken down as follows:

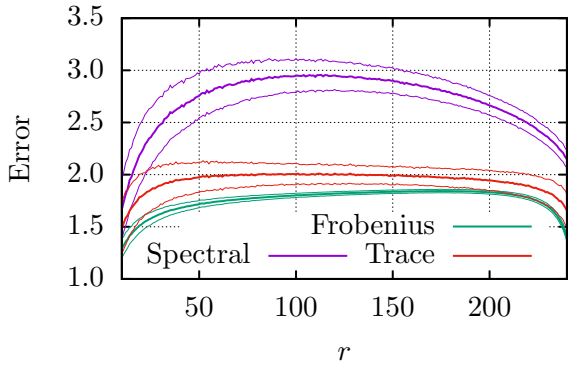
- Line 3 is $\mathcal{O}(n(r+l))$ for creating the random matrix Ω.
- Line 4 is $\mathcal{O}(mn(r+l))$ for the product $A\Omega$ and $\mathcal{O}\big(m(r+l)^2\big)$ for the QR decomposi-



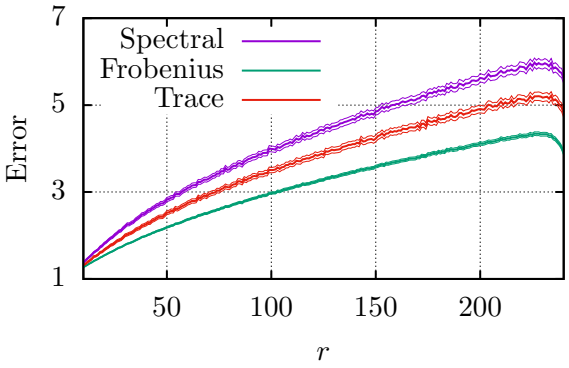
(a) Full rank.



(b) Rank-r.



(c) Algebraically decaying singular values ($\sigma_i = 10 \times i^{-1.5}$).



(d) Geometrically decaying singular values ($\sigma_i = 10 \times 0.9^{i-1}$).

Figure 1: Relative error of four types of matrix for the spectral, Frobenius, and trace norms. The thin lines represent the standard deviation in Figures 1a and 1c, the standard error in Figure 1d, and the error of the truncated SVD in Figure 1b. The parameters of the computations were $m = 500$, $n = 250$, $l = 5$, and $N = 10^3$.

tion[1].

- Line 5 is $\mathcal{O}\big(n(r+l)^2\big)$ for the SVD.

Therefore, the process is bottlenecked by the QR decomposition, and will scale as $\mathcal{O}(mnr)$, as opposed to $\mathcal{O}\big(mn^2\big)$ as in truncated SVD which first calculates the full SVD. Thus, we would expect randomized SVD to be $\mathcal{O}(n/r)$ faster, which for a large, redundant matrix can be quite advantageous.

We implement Algorithm 1 in Julia[2], and compare the relative error in the spectral, Frobenius, and trace norms for matrices of full rank, rank-r, algebraically decaying singular values, and geometrically decaying singular values[3]. The results of the computations can be seen in Figure 1.

We can see from Figure 1a that the relative error monotonically increases for all three norms—however, once $r+l$ approaches n the relative error quickly approaches one as the approximation becomes exact. Furthermore, even for a somewhat sizeable matrix the error is indeed $\mathcal{O}(1)$, and the standard deviation quite small. We find very different behaviour for rank-r matrices (Figure 1b). In this case we are approximating a rank-r matrix with a rank-r matrix, and so we expect to—mathematically—have a zero error. Of course computationally we are unlikely to obtain zero, and instead we find errors of $\mathcal{O}\big(10^{-11}\big)$. Randomized SVD performs considerably worse here (relatively) than truncated SVD, but, we are still able to approximate the matrix almost exactly. For matrices whose singular values decay algebraically (Figure 1c) we again find a different relation. The relative error quickly plateaus around three for the spectral norm, and two for the Frobenius and Trace norms. Since the singular values in this case decay slowly they carry a similar weight as their neighbours. Hence, the error is fairly constant for reasonably sized values of r. Finally, matrices with singular values that decay geometrically (Figure 1d) display similar behaviour as full rank matrices, but, with a negative concavity. As the singular values decay exponentially, the first few are exceedingly important. As r increases the chance of finding these initial singular values also increases, and so, the error begins to level off.

[1] As we shall see, Algorithm 1 is implemented in Julia which calls the C Householder QR decomposition [4].

[2] The code can be found at `github.com/bmetherall/Oxford/blob/master/Courses/Computational_Techniques/RandSVD.jl`.

[3] In the case of rank-r matrices we compare the absolute error instead since the approximation should be exact.

4 Conclusion

SVD has many applications in statistics and data science for analysis and compression. However, the SVD and truncated SVD can become quite expensive to compute for large matrices. Randomized SVD helps alleviate this problem as we saw in Section 3. The key observation in the case of all four types of matrices is that the relative error for each norm is $\mathcal{O}(1)$. This is very promising—for example, suppose we wish to approximate a full rank 500×250 matrix by a rank-100 matrix. Randomized SVD yields a matrix with an expected relative error less than 1.4, but, it can be computed approximately $2.5\times$ faster. For very large matrices with relatively low rank, as is common in data science, randomized SVD is much more effective than standard SVD.

References

[1] S. Axler, *Linear Algebra Done Right*. Undergraduate Texts in Mathematics, Springer, 2 ed., 1997.

[2] C. Eckart and G. Young, "The Approximation of One Matrix by Another of Lower Rank," *Psychometrika*, vol. 1, pp. 211–218, Sept 1936.

[3] L. Mirsky, "Symmetric Gauge Functions and Unitarily Invariant Norms," *The Quarterly Journal of Mathematics*, vol. 11, pp. 50–59, Jan 1960.

[4] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia Linear Algebra Documentation." Online, Retrieved Nov 2019. url: `docs.julialang.org/en/v1/stdlib/LinearAlgebra/`.