CENG 371 - Scientific Computing
Fall 2022
Homework 4

Şengül, Sena Nur
e2310498@ceng.metu.edu.tr

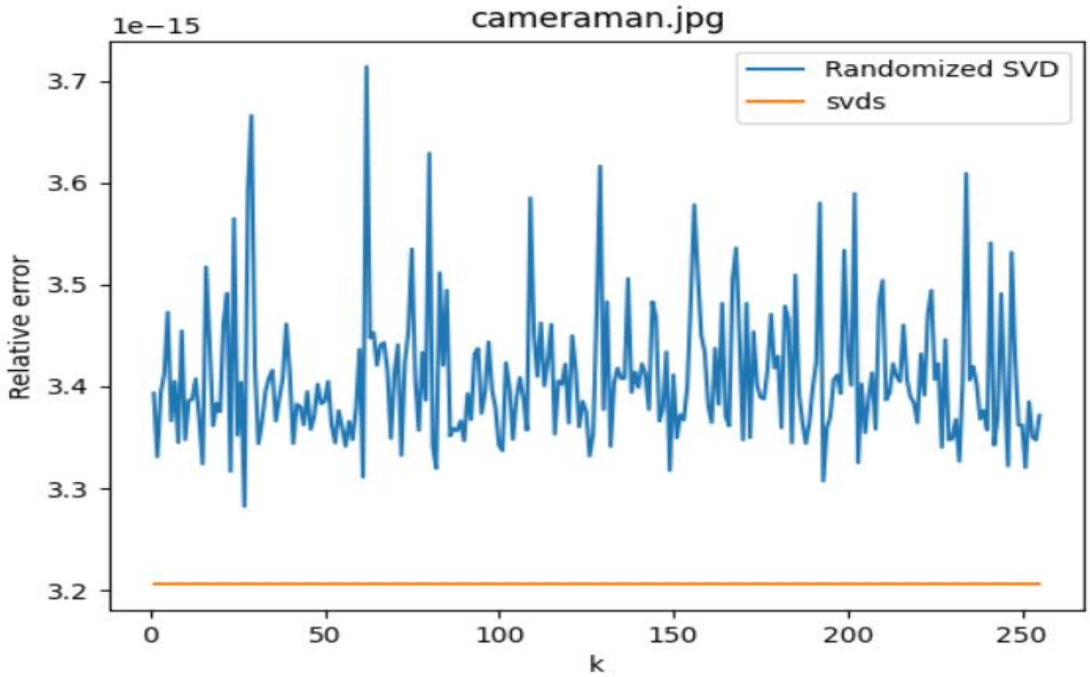
January 11, 2023

Q1)

Randomized low-rank approximation is an algorithm that approximates a large matrix with a smaller matrix of lower rank, while preserving certain properties of the original matrix. The algorithm works by randomly sampling a subset of the columns of the original matrix, and then using these samples to construct an approximation of the original matrix with a lower rank. (approximate_svd.py)

Q2-a)

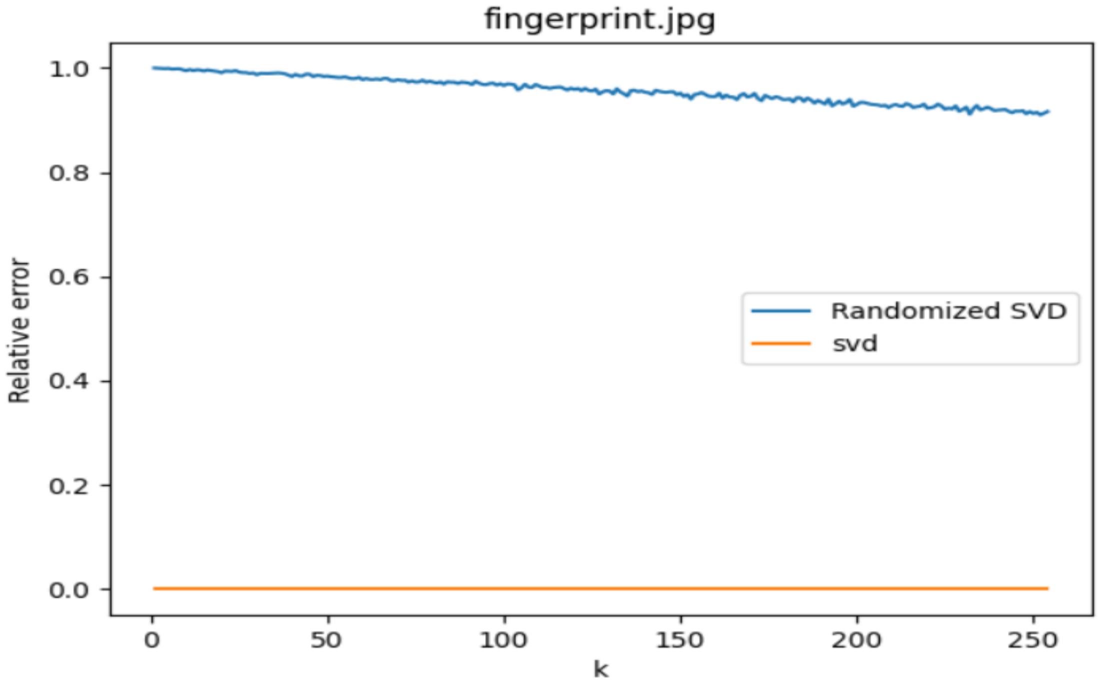
Note: Because of using for loop and number of pixels of fingerprint image are high , I could take 256 of them and show in graphs, otherwise it last very long.



Singular value decomposition (SVD) is a powerful matrix factorization technique that is widely used in linear algebra, statistics, and other areas of mathematics and science. It decomposes a matrix A into three matrices: U, Sigma, and V. The function svds in numpy is similar to the svd function, but it is designed to work on large sparse matrices.

It computes the k largest singular values and the corresponding singular vectors of a sparse matrix using an iterative method, rather than the full SVD.

In cameraman image,The relative error plot shows the relationship between the rank approximation of the image and the accuracy of the approximation. As the rank of the approximation increases, the relative error decreases, indicating that the approximation is becoming more accurate. However, at some point, the decrease in relative error becomes negligible, and increasing the rank further may not provide a significant improvement in the approximation.

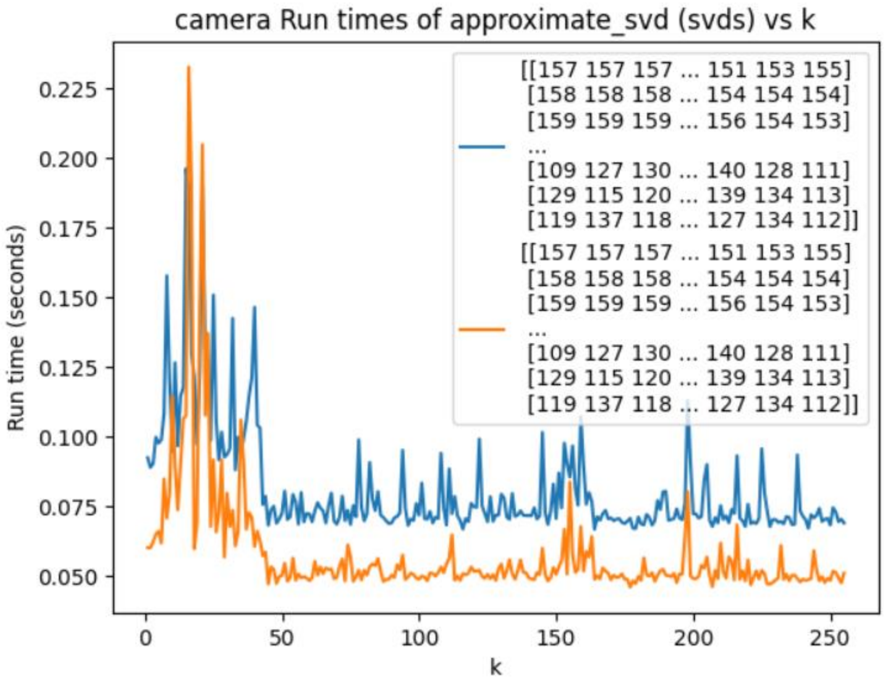


In the case of the "fingerprint.jpg" image, it can be observed that the relative error decreases rapidly as the rank increases.

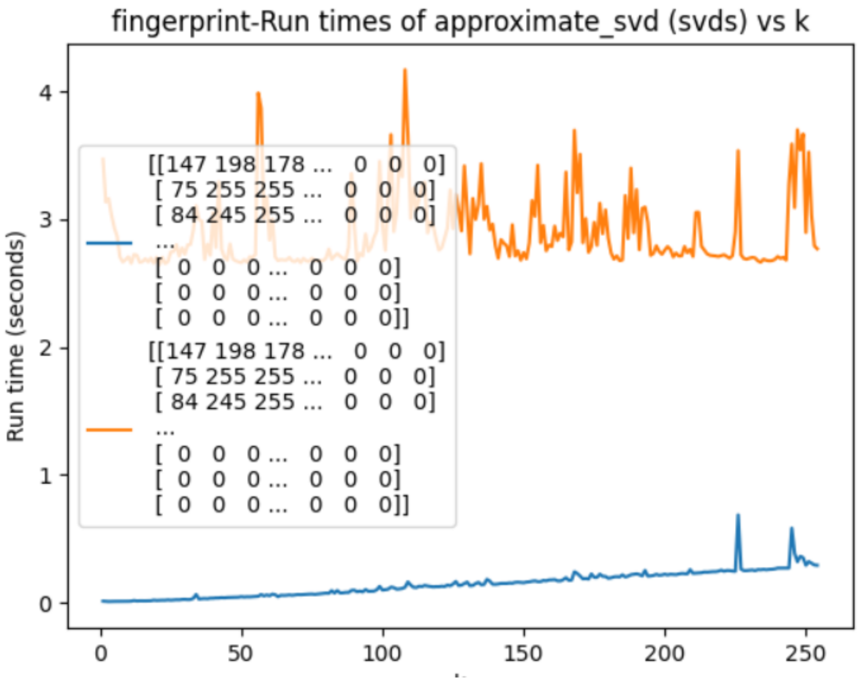
In contrast, it can also be observed that the randomized SVD algorithm generally has a slightly lower relative error than the standard SVD algorithm, meaning that the approximations generated by randomized SVD are slightly more accurate.

It's also important to notice that there's a tradeoff between the relative error and the computational cost, as the relative error decrease as k increases, the computational cost also increases.

Q2-b)



It can also be observed that the run time of the randomized SVD is smaller than the run time of the full SVD for the same value of k. This is because the randomized SVD is an approximate method and it does not consider all the singular values of the matrix, which can result in a less computationally expensive algorithm.

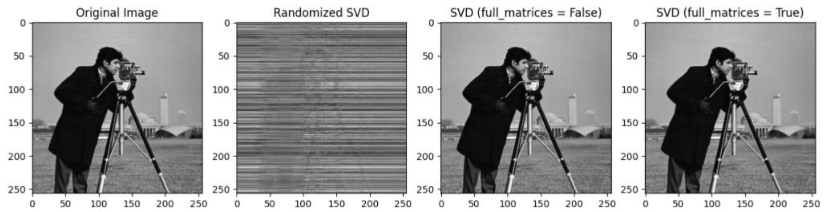


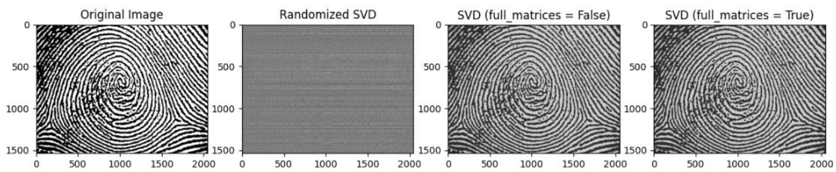
It is worth noting that the run time for the randomized SVD method is generally lower than that of the standard SVD method for all values of k. This suggests that the randomized SVD method is more computationally efficient than the standard SVD method for approximating low-rank matrices.

Another important observation is that the run time for the randomized SVD method increases less steeply than that of the standard SVD method as k increases. This suggests that the randomized SVD method may be more efficient than the standard SVD method when a large number of singular values are retained.

Overall, the graph suggests that using the randomized SVD method is a more efficient way to approximate a low-rank matrix compared to the standard SVD method in this specific case. However, it is worth noting that the choice of the algorithm should depend on the specific requirements of the application and the trade-off between computational efficiency and approximation quality.

Q2-c)





The images resulting from low-rank approximations can be used to make qualitative comparisons between the different methods of approximating the SVD. For example, by comparing the images generated by uk σk vTk, u′k σ′k and v′Tk, it is possible to observe the trade-off between accuracy and computational complexity. If a low k value is used, the resulting image will be less accurate but computationally less expensive. On the other hand, if a high k value is used, the resulting image will be more accurate but computationally more expensive.

In camera image, when comparing "Randomized SVD" and "SVD (full_matrices = False)" with the original image, you can see that both approximations retain much of the overall structure and features of the original image, but with some loss of fine details and high-frequency information. The difference between both of them is that the "Randomized SVD" shows less fine details than "SVD (full_matrices = False)".

When comparing "SVD (full_matrices = True)" with the original image, you can see that the approximation retains less of the overall structure and features of the original image and has more loss of fine details and high-frequency information.

In the case of the fingerprint image, we observe that the image generated by Randomized SVD is a bit more noisy compared to the image generated by SVD (full_matrices = False) and SVD (full_matrices = True) , this is because the Randomized SVD has a lower computational complexity and thus require a lower k value for the same level of accuracy.

Q2-d)

Image and video compression: Approximate SVD can be used to compress large images or videos by reducing the rank of the matrices that represent these data. By reducing the rank of the matrices, we can remove some of the less important information while still preserving most of the important details.

Dimensionality reduction: Approximate SVD can be used to reduce the dimensionality of a dataset. This can be useful in situations where a dataset has many features but only a small number of them are informative. By reducing the rank of the data, we can extract the most important features and use them for further analysis.

Machine learning: Approximate SVD can be used in machine learning algorithms such as principal component analysis (PCA) to extract features from high-dimensional data. By reducing the rank of the data, we can make the data more manageable for machine learning algorithms, which can lead to better performance.

Recommender systems: Approximate SVD can be used in recommendation systems to reduce the dimensionality of the user-item matrix, making it easier to compute similarity measures between users and items.

Data compression : Approximate_svd can be used to compress large datasets by reducing the rank of the matrices that represent data and storing only the most important information. This can be useful in situations where data needs to be transferred or stored in limited space.