CMDA 3605 **Jung H Choi**

Term Project
SVD and Its Application to Dynamical Systems

Abstract

The goal of this project is to employ your Difference and Differential Equations, Linear Algebra and Matlab skills for applications in Dynamical Systems. At the end, these skills will be combined to construct dynamical systems directly from data (without access to original mathematical equations) and to reduce the number of equations in a dynamical system. The main tool in achieving this goal would be an important matrix decomposition, called the Singular Value Decomposition (SVD).

Some instructions on the structure of the term project

Discussion below explains the required concepts in detail. Through out the text, several problems are listed that you need to answer. However, your term project, once completed, should be a coherent report. Therefore do **NOT** simply answer the problems. Follow the pattern of the document below. You may use the same language or use other resources; but in both cases cite the appropriate source. Then, when you reach a certain problem, such as Problem 2.1, 2.2 etc, include its answer in the corresponding location and continue your discussion. Your criterion should be that your term project needs to be a coherent, well-written report; rather than a collection of solutions to specific problems. You are required to type your term project using your favorite editing software. Term projects that do not offer a coherent discussion (i.e. simply list solutions to the specific problems like a homework) and/or that are hand written will loose at least 20%. A printed hardcopy should be turned in, in addition to uploading it into Canvas. For the numerical problems, in addition to including your results and/or figures, attach your Matlab code as an Appendix. Also, as in the homework assignments, the code should also be uploaded into Canvas.

1 PART I: The singular value decomposition (SVD)

The **Singular Value Decomposition**(SVD) is a decomposition into the form

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\mathbf{S}$ is an mxn diagonal matrix whose elements are the singular values of the original matrix. If A is a complex matrix, then there always exists such a decomposition with positive singular values[3]. We will further look in to details of what SVD does then use in the real-life application.

1.1 Problem 1.1

Problem 1.1 (a) Let $\mathbf{A} \in \mathbb{R}^{n\times m}$, show that $\mathbf{A}$ can be written as

$$\mathbf{A} = \sum_{i=1}^{m} \sigma_i \mathbf{u}_i \mathbf{v}_i^T.$$

We are looking for the sum of the first m matrices.

$$\mathbf{A} = [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_m] \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_m \end{bmatrix} [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \cdots \quad \mathbf{v}_m]^T \tag{1}$$

$$= [\mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_m] \begin{bmatrix} \sigma_1 \mathbf{v}_1^T \\ \sigma_2 \mathbf{v}_2^T \\ \vdots \\ \sigma_m \mathbf{v}_m^T \end{bmatrix}$$

$$= \sigma_1 \mathbf{u}_1 \mathbf{v}_1^T + \ldots \sigma_2 \mathbf{u}_2 \mathbf{v}_2^T + \ldots \sigma_3 \mathbf{u}_3 \mathbf{v}_3^T + \ldots + \sigma_m \mathbf{u}_m \mathbf{v}_m^T = \sum_{i=1}^{m} \sigma_i u_i v_i^T$$

(b) Let $\mathbf{A}_k \in \mathbb{R}^{n \times m}$ (the best rank-k approximation to $\mathbf{A}$) be obtained as in (**??**). Show that

$$\|\mathbf{A} - \mathbf{A}_k\|_2 = \sigma_{k+1}.$$

$$= \| \sum_{i=1}^{m} \sigma_i u_i v_i^T - \sum_{i=1}^{k} \sigma_i u_i v_i^T \|_2 = \| \sum_{i=k+1}^{m} \sigma_i u_i v_i^T \|_2$$

The 2-norm of matrix is equivalent to square root of the largest eigenvalue of $\mathbf{A}^T\mathbf{A}$. In order to find $\|\mathbf{A}\|_2$,

$$\| \sum_{i=k+1}^{m} \sigma_i u_i v_i^T \|_2 = \sqrt{max_i \lambda_i(\mathbf{A}^T\mathbf{A})} = \sqrt{\lambda}; eigenvalues$$

$$= \sqrt{((\underbrace{\sum_{i=k+1}^{m} \sigma_i u_i v_i^T)^T}_{\mathbf{A}^T} (\underbrace{\sum_{i=k+1}^{m} \sigma_i u_i v_i^T}_{\mathbf{A}}))}$$

$$= \sqrt{(\sigma_{k+1}\mathbf{v}_{k+1}\mathbf{u}_{k+1}^T + \ldots + \sigma_m \mathbf{v}_m \mathbf{u}_m^T)(\sigma_{k+1}\mathbf{u}_{k+1}\mathbf{v}_{k+1}^T + \ldots + \sigma_m \mathbf{u}_m \mathbf{v}_m^T)}$$

$$= \begin{bmatrix} \sigma_{k+1}^2 & & & \\ & \sigma_{k+2}^2 & & \\ & & \ddots & \\ & & & \sigma_m^2 \end{bmatrix} = \sigma_{k+1}, \sigma_{k+2}, \ldots, \sigma_m$$

The singular value decreases $\sigma_{k+1} > \sigma_{k+2} > \ldots > \sigma_m$ as k increases. We have solved the rank-k approximation of A.

1.2 Computing SVD by-hand for small matrices

For this section we will find how to build SVD for small matrices. This focuses on finding singular values, left and right singular vectors and how these can be related.

Problem 1.2 Given $\mathbf{A} \in \mathbb{R}^{n \times m}$, let $\mathbf{A} = \mathbf{USV}^T$ be the SVD of $\mathbf{A}$ where

$$\mathbf{S} = \operatorname{diag}(\sigma_1, \cdots, \sigma_n), \quad \mathbf{U} = [u_1 \ \ ... \ \ u_n], \quad \mathbf{V} = [v_1 \ \ ... \ \ v_n] \quad \text{and} \quad \mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}_n. \tag{2}$$

$$\begin{aligned}
\mathbf{A} &= \mathbf{USV}^T, && (\mathbf{A}^T = \mathbf{VS}^T\mathbf{U}^T) \\
\mathbf{AA}^T &= \mathbf{USV}^T(\mathbf{USV}^T)^T, && \textit{Left singular vector } u_j \textit{ are eigenvectors of } \mathbf{AA}^T \\
&= \mathbf{USS}^T\mathbf{U}^T \\
&= \mathbf{US}^2\mathbf{U}^T, && \textit{since } \mathbf{S} \textit{ is diagonal}
\end{aligned} \tag{3}$$

$$\begin{aligned}
\mathbf{A} &= \mathbf{USV}^T, && (\mathbf{A}^T = \mathbf{VS}^T\mathbf{U}^T) \\
\mathbf{A}^T\mathbf{A} &= \mathbf{VS}^T\mathbf{U}^T(\mathbf{USV}^T), && \textit{Right singular vector } v_j \textit{ are eigenvectors of } \mathbf{A}^T\mathbf{A} \\
&= \mathbf{VS}^T\mathbf{SV}^T \\
&= \mathbf{VS}^2\mathbf{V}^T, && \textit{since } \mathbf{S} \textit{ is diagonal}
\end{aligned} \tag{4}$$

Each $\mathbf{S}^2$ (σ_i^2) is an eigenvalue of A^TA and AA^T. When we write down singular values in the following order,

$$\sigma_1 \geq \sigma_2 \geq ...\sigma_r > 0$$

, each of them has rank 1 then the singular decomposition gives A in r rank 1 pieces in order of importance as mentioned in 1.1. We are given that $\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}$ then we can rewrite both left and right eigenvectors as $\mathbf{US}^2\mathbf{U}^{-1}$ and $\mathbf{VS}^2\mathbf{V}^{-1}$ since ($\mathbf{U}^T = \mathbf{U}^{-1}$ and $\mathbf{V}^T = \mathbf{V}^{-1}$) These matrices are in the form of $X\Lambda X^{-1}$=A where Λ is the diagonal matrix with X = U = V eigenvectors of the matrix A. Then we have V and U as eigenvectors of A^TA and AA^T along the diagonal matrix of $\mathbf{S}^2$.

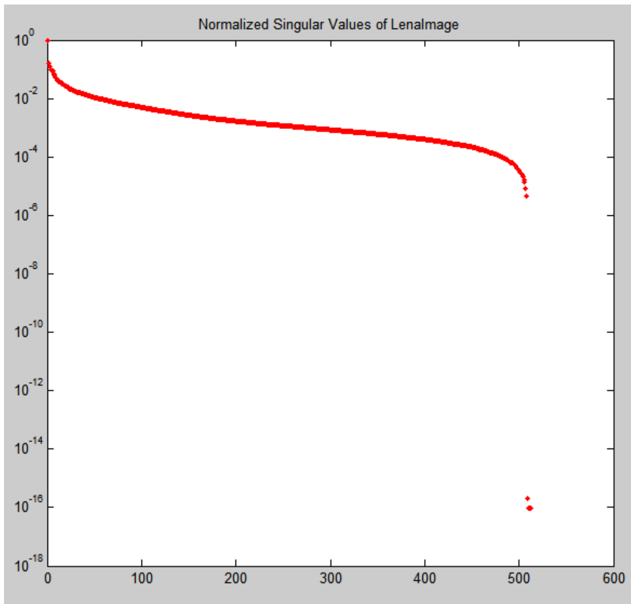
1.3 Image Approximation by Singular Value Decomposition

Problem 1.4

Approximation of the Lena image using **Matlab**.

(a) plot the **normalized** singular values of A on a logarithmic scale.

This plot shows the normalized singular values of the Lena image. Ignoring the obvious outliers, we can



see the data up to hundred are within the 10^{-2} then majority of the data are within the 10^{-4} orders of magnitude.

Now we will find the optimal approximants having relative error less than 10^{-1}, 10^{-2}, $5{*}10^{-3}$ and plot them.

(b) Compute the optimal approximants in the 2-norm having a **relative error** less than 10^{-1},10^{-2}, and 5 x 10^{-3}.

There is not a significant amount of a difference between the original image and the image shown on Figure 4. Having much lower rank is big advantage since the original image takes up about 5 times more than the one at rank 4.

Figure 1: Original image.

Figure 2: Rank k = 4, Relative Error = 0.0909

Figure 3: Rank k = 4, Relative Error = 0.0100

Figure 4: Rank k = 4, Relative Error = 0.0050

1.4 Face Recognition by Eigenfaces

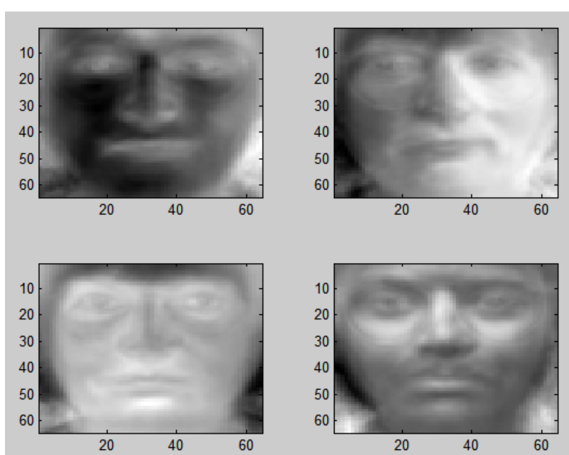
In this section, we will now apply SVD onto a large sets of Yaleface database. One major application of SVD for this purpose is the image reconstruction based on a database. Since SVD reveals the optimal information for approximation, it would be best tool for this section.

Problem 1.5
Average face is very simple to create. We just take every image, Y_i and add them up one by one then divide them by the total length, **n**.

(a) Construct the average face for this database and plot it.

(b) Construct the database matrix D and compute its SVD. Plot the leading four eigenfaces. How do these eigenfaces look like?

The database matrix, D is formed by subtracting the average face from every image. Then we will examine the matrix by taking the SVD form. These four leading eigenfaces are the most dominant information



within the database we have. Although, the images aren't so clear, they are constructed based on common facial information that we gathered from the database.

(C) Download image_to_reconstruct Reconstruct: F1, f2, and F3.

We can now use the information we already have to reconstruct other matrices that are not in our data base. Reconstructing the original images (F1, F2, F3) with the eigenfaces of the other individuals that we have results in the following.

Figure 5: The original image before is on the left and after reconstruction is to the right

This is the power of SVD. Once can improve the quality of these reconstruction further by expanding the database by including richer images. One can apply these techniques to reconstruct images with missing parts.

References

[1] G. Golub and C. van Loan, *Matrix Comptutations*, 3rd Edition, JHU Press, 2012.

[2] N. Muller, L. Magaia, and B.M. Herbst, *Singular value decomposition, eigenfaces, and 3D reconstructions*, SIAM review, Vol. 46, No. 3, pp: 518–545, 2004. http://mathworld.wolfram.com/SingularValueDecomposition.html

[3] wolfram, *SingularValueDecomposition.html* mathworld.wolfram.com/SingularValueDecomposition.html.