Overview of Topics

(1) Singular Value Decomposition (SVD)

* Dimensionality reduction
* Data Analysis
* Machine Learning

(2) Today:

(A) What is the SVD

(B) $X = U \Sigma V^*$

(C) Image Compression.

Chapter 1

Singular Value Decomposition (SVD) and Principal Components Analysis (PCA)

The singular value decomposition (SVD) is among the most important matrix factorizations of the computational era, providing a foundation for nearly all of the data methods discussed in this book. In particular, the SVD provides a numerically stable matrix decomposition that can be used for a variety of purposes. We will use the SVD to obtain low-rank approximations to matrices and to perform pseudo-inverses of non-square matrices to find the least-squares and minimum norm solutions of a matrix system of equations $\mathbf{Ax} = \mathbf{b}$. Another important use of the SVD is in the principal components analysis (PCA), whereby a large, high-dimensional data set is decomposed into its most statistically descriptive factors. SVD/PCA has been applied to a tremendous variety of problems in science and engineering, making it the most generally useful computational tool after the fast Fourier transform (FFT).

1.1 Overview (Big Picture)

Here we introduce the singular value decomposition (SVD) and give an overview of some of the intuitive motivating examples. Detailed mathematical properties are discussed in the following sections.

1.1.1 Intuition and applications

In this chapter, we will develop an intuition for how to apply the SVD by demonstrating its use on a number of motivating examples. These methods will provide a foundation for many other techniques developed in this book, including classification methods in Chapter 2, the proper orthogonal decomposition (POD) in Chapter 3, and the dynamic mode decomposition (DMD) in Chapter 4.

High-dimensionality is a common challenge in processing data from large complex systems. These systems may involve large measured data sets including audio, image, or video data. The data may also be generated from a complex physical system, such as neural recordings from the mammalian cortex, or fluid velocity measurements from a simulation or experiment. In any case, it is observed that most data from naturally occurring systems exhibits dominant patterns of activity, which may be characterized by a low-dimensional attractor or manifold [26, 25].

As an example, consider images, which typically contain a large number of measurements (pixels), and are therefore elements of a high-dimensional vector space. However, most images are highly compressible, meaning that the relevant information may be represented in a much lower dimensional subspace. The compressibility of images will be discussed in depth throughout this book. Complex fluid systems, such as the turbulent wake behind a vehicle or the Earth's atmosphere also provide compelling examples the low-dimensional structure underlying a high-dimensional state-space. Although high-fidelity fluid simulations typically require at least millions or billions of degrees of freedom, there are often dominant coherent structures in the flow, such as periodic vortex shedding behind vehicles or hurricanes in the weather.

The SVD provides a systematic way to determine the dominant patterns underlying a high-dimensional system, providing a low-rank approximation to high-dimensional data. This technique is *data-driven* in that patterns are discovered purely from the data, without the addition of any expert knowledge or intuition. The SVD may be thought of as a numerically stable computation that provides a hierarchical representation of the data in terms of a new coordinate system defined by dominant correlations within the data.

The SVD has many powerful applications beyond dimensionality reduction of high-dimensional data. It will be useful in computing the pseudo-inverse of non-square matrices, providing solutions to underdetermined or overdetermined matrix equations that are either minimum norm solutions, or solutions that minimize the sum-squared error. We will also use the SVD in a principled approach to de-noising data sets. The SVD is also important to characterize the input and output geometry of a linear map between vector spaces. These applications will all be explored in this chapter, providing an intuition for matrices and high-dimensional data.

1.1.2 Definition

Generally, we are interested in analyzing a large data set $\mathbf{X}$:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_m \\ | & | & & | \end{bmatrix}. \tag{1.1a}$$

The columns $\mathbf{x}_k \in \mathbb{C}^n$ may be measurements from simulations or experiments. For example, columns may represent images that have been reshaped into column vectors with as many elements as pixels in the image. The column vectors may also represent the state of a physical system that is evolving in time, such as the fluid velocity at each point in a discretized simulation or at each measurement location in a wind-tunnel experiment.

The index k is a label indicating the k^{th} distinct set of measurements; for many of the examples in this book $\mathbf{X}$ will consist of a *time-series* of data, and $\mathbf{x}_k = \mathbf{x}(k\Delta t)$. Often the *state-dimension* n is very large, on the order of millions or billions in the case of fluid systems. The columns are often called *snapshots*, and m is the number of snapshots in $\mathbf{X}$. For many systems $n \gg m$, resulting in a *tall-skinny* matrix, as opposed to a *short-fat* matrix when $n \ll m$.

The SVD is a unique matrix decomposition that exists for every complex valued matrix $\mathbf{X} \in \mathbb{C}^{n\times m}$:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* \tag{1.2a}$$

where $\mathbf{U} \in \mathbb{C}^{n\times n}$ and $\mathbf{V} \in \mathbb{C}^{m\times m}$ are *unitary* matrices[1] and $\mathbf{\Sigma} \in \mathbb{C}^{n\times m}$ is a matrix with non-negative entries on the diagonal and zeros off the diagonal. Here * denotes the complex conjugate transpose[2]. As we will discover throughout this chapter, the condition that $\mathbf{U}$ and $\mathbf{V}$ are unitary is extremely powerful.

The matrix $\mathbf{\Sigma}$ has at most m non-zero elements on the diagonal, and may therefore be written as $\mathbf{\Sigma} = \begin{bmatrix} \hat{\mathbf{\Sigma}} \\ \mathbf{0} \end{bmatrix}$. Therefore, it is possible to *exactly* represent $\mathbf{X}$ using the *reduced* SVD:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* = \begin{bmatrix} \hat{\mathbf{U}} & \hat{\mathbf{U}}^\perp \end{bmatrix} \begin{bmatrix} \hat{\mathbf{\Sigma}} \\ \mathbf{0} \end{bmatrix} \mathbf{V}^* = \hat{\mathbf{U}}\hat{\mathbf{\Sigma}}\mathbf{V}^*. \tag{1.3}$$

The full and reduced SVD are shown in Fig. 1.1. Here $\hat{\mathbf{U}}^\perp$ is complementary to $\hat{\mathbf{U}}$.

The columns of $\mathbf{U}$ are called *left singular vectors* of $\mathbf{X}$ and the columns of $\mathbf{V}$ are *right singular vectors*. The diagonal elements of $\hat{\mathbf{\Sigma}} \in \mathbb{C}^{m\times m}$ are called *singular values* and the are ordered from largest to smallest.

In Matlab, the computing the SVD is straightforward:

```
>>[U,S,V] = svd(X);   % Singular Value Decomposition
```

For non-square matrices $\mathbf{X}$, the reduced SVD may be computed more efficiently using:

```
>>[Uhat,Shat,V] = svd(X,'econ');   % economy sized SVD
```

[1] A square matrix $\mathbf{U}$ is unitary if $\mathbf{U}\mathbf{U}^* = \mathbf{U}^*\mathbf{U} = \mathbb{I}$.

[2] For real-valued matrices, this is the same as the regular transpose $\mathbf{X}^T$.

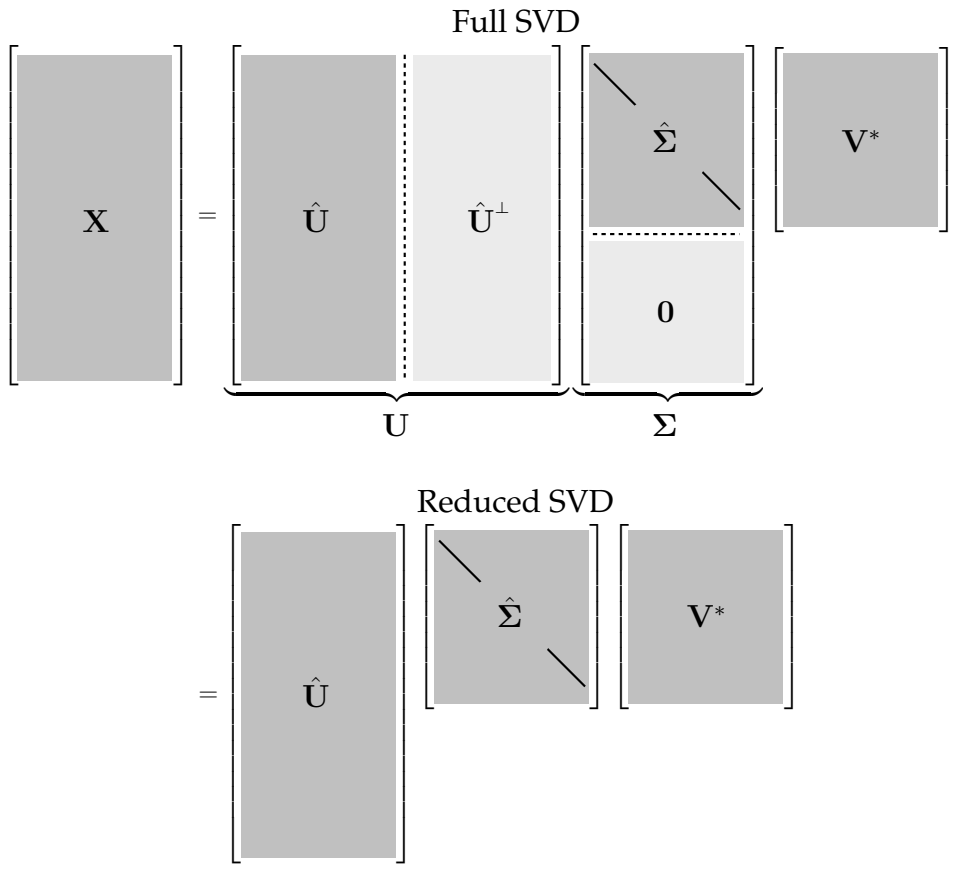

Figure 1.1: Schematic of matrices involved in the full SVD and the reduced SVD.

1.1.3 Historical Perspective

The SVD has a long and rich history, ranging from early work developing the theoretical foundations to modern work on computational stability and efficiency. There is an excellent historical review by Stewart [50], which provides context and many important details. The review focuses on the early theoretical work of Beltrami and Jordan (1873), Sylvester (1889), Schmidt (1907), and Weyl (1912). It also discusses more recent work, including the seminar computational work of Golub and collaborators [20, 19]. In addition, there are many excellent chapters on the SVD in modern books [52, 2, 33].

1.1.4 Uses in This Book and Assumptions of the Reader

The SVD is the basis for many related techniques in dimensionality reduction used to obtain reduced order models (ROMs). These methods indlude principal components analysis (PCA) in statistics [40, 27, 28], the Karhunen–Loève transform (KLT) [30, 36], empirical orthogonal functions (EOFs) in climate [37], the proper orthogonal decomposition (POD) in fluid dynamics [25], and canonical correlation analysis (CCA) [10]. Although developed independently in a range of diverse fields, many of these methods only differ in

how the data is collected and pre-processed. There is an excellent discussion about the relationship between the SVD, the KLT and PCA [17].

The SVD is also widely used in system identification and control theory to obtain reduced order models that are balanced in the sense that states are hierarchically ordered in terms of their ability to be observed by measurements and controlled by actuation [39].

For this chapter, we assume that the reader is familiar with linear algebra with some experience in computation and numerics. For review, there are number of excellent books on numerical linear algebra, with discussions on the SVD [52, 2, 33].

1.2 Matrix Approximation

Perhaps the most useful and defining property of the SVD is that it provides an *optimal* low-rank approximation to a matrix $\mathbf{X}$. In fact, the SVD provides a *hierarchy* of low-rank approximations, since a rank-r approximation is obtained by keeping the leading r singular values and vectors, and discarding the rest.

Schmidt (of Gram-Schmidt) generalized the SVD to function spaces and developed an approximation theorem, establishing truncated SVD as the optimal low rank approximation of the underlying matrix $\mathbf{X}$ [47]. Schmidt's approximation theorem was rediscovered by Eckart and Young [13], and is sometimes referred to as the Eckart-Young theorem.

Theorem 1 (Eckart-Young [13]) *The optimal rank-r approximation to $\mathbf{X}$, in an L_2 sense, is given by the rank-r SVD truncation $\tilde{\mathbf{X}}$:*

$$\underset{\tilde{\mathbf{X}}}{\operatorname{argmin}} \|\mathbf{X} - \tilde{\mathbf{X}}\|_2 = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^*. \tag{1.4}$$

Here, $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ denote the leading r columns of $\mathbf{U}$ and $\mathbf{V}$, and $\tilde{\mathbf{\Sigma}}$ contains the leading $r \times r$ sub-block of $\mathbf{\Sigma}$.

Here, we establish the notation that a truncated SVD basis (and the resulting approximated matrix $\tilde{\mathbf{X}}$) will be denoted by $\tilde{\mathbf{X}} = \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^*$. Because $\mathbf{\Sigma}$ is diagonal, the rank-r SVD approximation is given by the sum of r distinct rank-1 matrices:

$$\tilde{\mathbf{X}} = \sum_{k=1}^{r} \sigma_k \mathbf{u}_k \mathbf{v}_k^* = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^* + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^* + \cdots + \sigma_r \mathbf{u}_r \mathbf{v}_r^*. \tag{1.5}$$

This is the so-called *dyadic* summation. For a given rank r, there is no better approximation for $\mathbf{X}$, in the L_2 sense, than the truncated SVD approximation $\tilde{\mathbf{X}}$.

This is an extremely important property of the SVD, and we will return to it many times. There are numerous examples of data sets that contain high-dimensional measurements, resulting in a large data matrix $\mathbf{X}$. However, there are often dominant low-dimensional patterns in the data, and the truncated SVD basis $\tilde{\mathbf{X}}$ provides a coordinate transformation from the high-dimensional measurement space into a low-dimensional pattern space. This has the benefit of *reducing* the size and dimension of large data sets, yielding a tractable basis for visualization and analysis. Finally, many systems considered in this text are *dynamic*, and the SVD basis provides a hierarchy of modes that characterize the observed attractor, on which we may project a low-dimensional dynamical system.

1.2.1 Truncation

The truncated SVD is illustrated in Fig. 1.2, with $\tilde{\mathbf{U}}, \tilde{\mathbf{\Sigma}}$ and $\tilde{\mathbf{V}}$ denoting the truncated matrices. If $\mathbf{X}$ does not have full row rank, then some of the singular values in $\hat{\mathbf{\Sigma}}$ may be zero, and the truncated SVD may still be exact. However, in general, for truncation values r that are smaller than the row rank of $\mathbf{X}$, the truncated SVD only approximates $\mathbf{X}$:

$$\mathbf{X} \approx \tilde{\mathbf{U}}\tilde{\mathbf{\Sigma}}\tilde{\mathbf{V}}^*. \tag{1.6}$$

There are numerous choices for the truncation rank r, and they are discussed in Sec. 1.7. If we choose the truncation value to be the rank of $\mathbf{X}$, so that $\hat{\mathbf{\Sigma}}_{\text{rem}}$ is zero and $\tilde{\mathbf{\Sigma}}$ contains non-zero singular values, then we may refer to $\tilde{\mathbf{U}}, \tilde{\mathbf{\Sigma}}, \tilde{\mathbf{V}}$ as $\hat{\mathbf{U}}_1, \hat{\mathbf{\Sigma}}_1, \hat{\mathbf{V}}_1$.

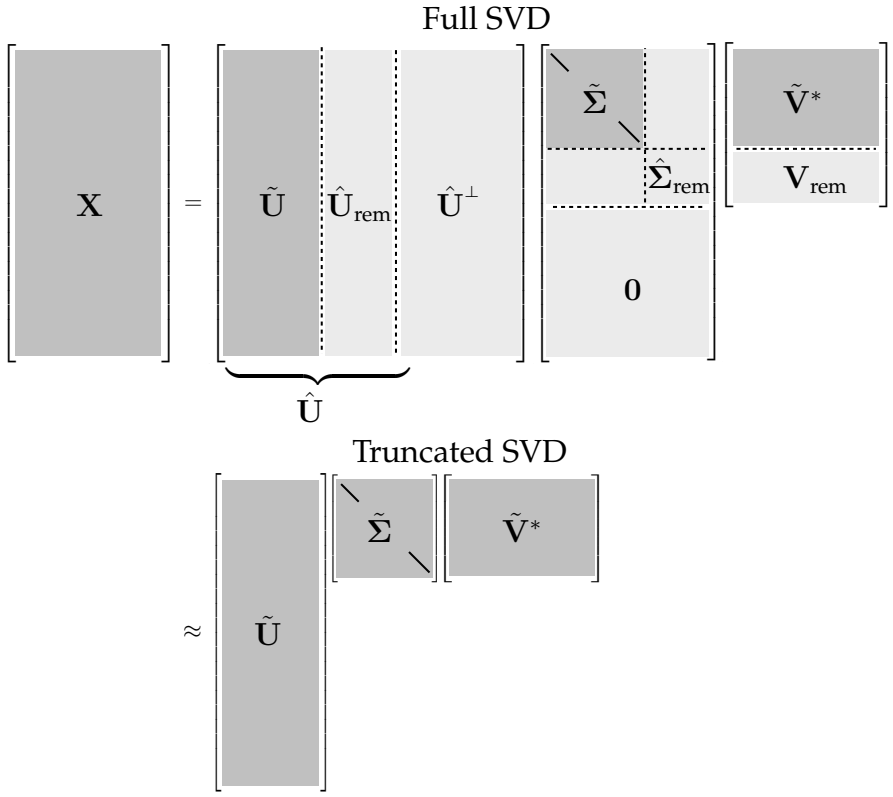

Figure 1.2: Schematic of truncated SVD. The subscript '$_{\text{rem}}$' denotes the remainder of $\hat{\mathbf{U}}, \hat{\mathbf{\Sigma}}$ or $\mathbf{V}$ after truncation.

1.2.2 Example: Image Compression

We demonstrate the idea of matrix approximation with a simple example: image compression. A recurring theme throughout this book is that large data sets often contain underlying patterns that facilitate low-rank representations. Natural images present a simple and intuitive example of this inherent *compressibility*. A grayscale image may be thought of as a real-valued matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$, where n and m are the number of pixels in the vertical and horizontal direction, respectively[3]. Depending on the basis of representation (pixel-space, Fourier frequency domain, SVD transform coordinates), images may have very compact approximations.

Consider the image of Mordecai the snow dog in Fig. 1.3. This image has 2000×1500 pixels. It is possible to take the SVD of this image and plot the diagonal singular values, as in Fig. 1.4. Figure 1.3 shows the approximate matrix $\tilde{\mathbf{X}}$ for various truncation values r. By $r = 100$, the reconstructed image is quite accurate, and the singular values account

[3]It is not uncommon for image size to be specified as horizontal by vertical, i.e. $\mathbf{X}^T \in \mathbb{R}^{m \times n}$, although we stick with vertical by horizontal to be consistent with generic matrix notation.

for almost 80% of the image variance. The SVD truncation results in a compression of the original image, since only the first 100 columns of $\mathbf{U}$ and $\mathbf{V}$, along with the first 100 diagonal elements of $\mathbf{\Sigma}$, must be stored.

First, we load the image of the dog and compute the SVD, as in Code 1.1. Next, we compute the approximate matrix using the truncated SVD for various ranks ($r = 5, 20,$ and 100) in Code 1.2. Finally, we plot the singular values and cumulative energy in Fig. 1.4 using Code 1.3.

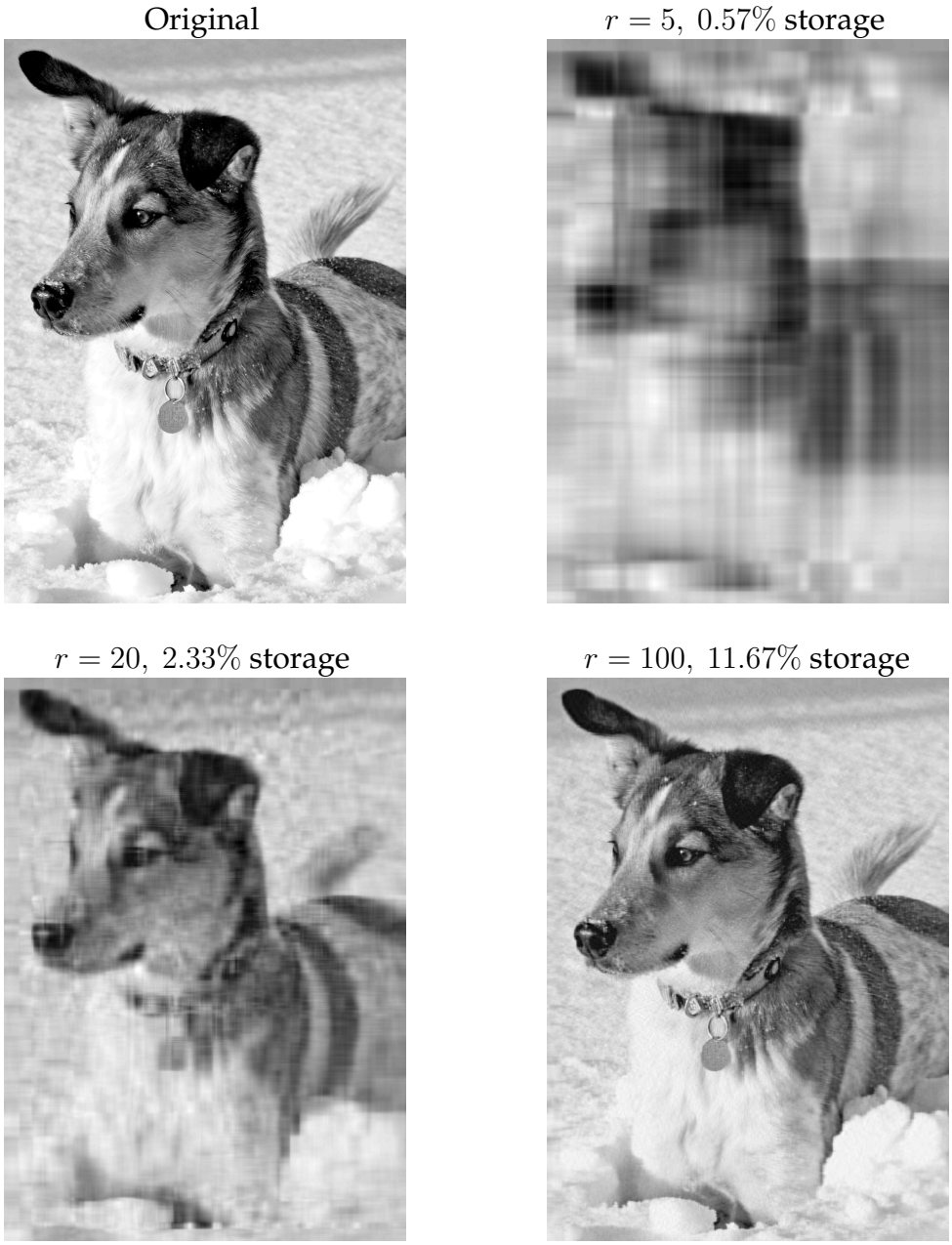

Figure 1.3: Image compression of Mordecai the snow dog, truncating the SVD at various ranks r. Original image resolution is 2000×1500. Generated by Codes 1.1 & 1.2.

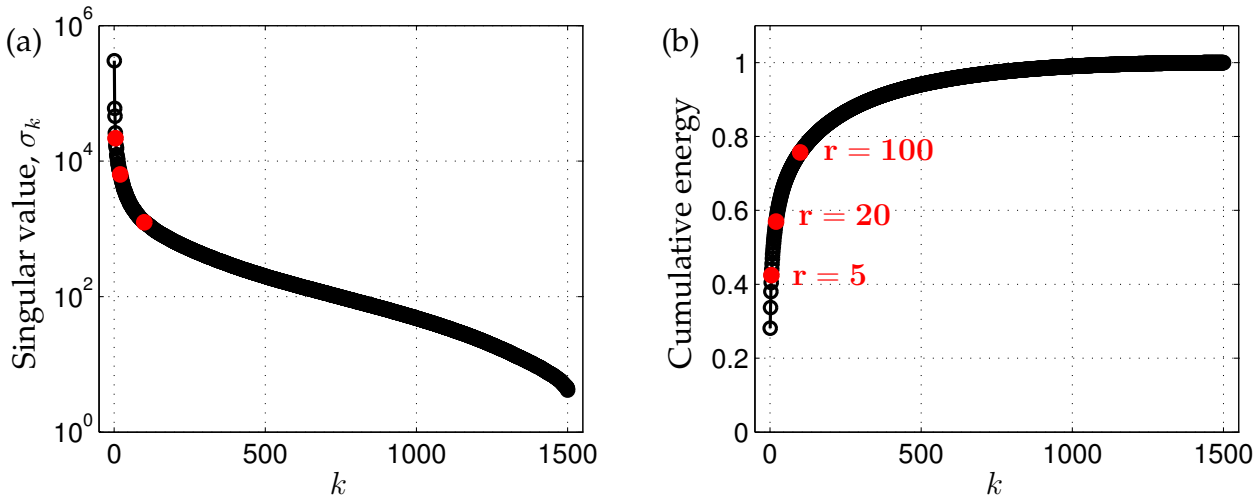

Figure 1.4: Singular values σ_r (a) and cumulative energy contained in the first r modes (b). Generated by Code 1.3.

Code 1.1: Load image and take the SVD.

```
A=imread('../DATA/dog.jpg');
X=double(rgb2gray(A));  % convert RBG to gray, 256 bit to double.
nx = size(X,1); ny = size(X,2);

[U,S,V] = svd(X);

figure, subplot(2,2,1)
imagesc(X), axis off, colormap gray
title('Original')
```

Code 1.2: Approximate image using truncated SVD for $r = 5, 20$ and 100. (Fig. 1.3)

```
plotind = 2;
for r=[5 20 100];  % truncation value
    Xapprox = U(:,1:r)*S(1:r,1:r)*V(:,1:r)';  % approximate image
    subplot(2,2,plotind), plotind = plotind + 1;
    imagesc(Xapprox), axis off
    title(['r=',num2str(r,'%d'),', ',num2str(100*r*(nx+ny)/(nx*ny),'
       %2.2f'),'% storage']);
end
set(gcf,'Position',[100 100 550 400])
```

Code 1.3: Plot the singular values and cumulative energy. (Fig. 1.4)

```
figure, subplot(1,2,1)
semilogy(diag(S),'k','LineWidth',1.2), grid on
xlabel('r')
ylabel('Singular value, \sigma_r')
xlim([-50 1550])
```

```
subplot(1,2,2)
plot(cumsum(diag(S))/sum(diag(S)),'k','LineWidth',1.2), grid on
xlabel('r')
ylabel('Cumulative Energy')
xlim([-50 1550])
ylim([0 1.1])
set(gcf,'Position',[100 100 550 240])
```

1.2.3 Aside on Compressibility and the Vastness of Image Space

It is important to note that the compressibility of images is related to the overwhelming vastness of image space. For even a simple 20×20 pixel black and white image, there are 2^{400} distinct possible images, which is larger than the number of nucleons in the known universe. The number of images is considerably more staggering for higher resolution images with greater color depth.

In the space of one megapixel images (i.e., 1000×1000 pixels), there is an image of us each being born, of me typing this sentence, and of you reading it. However vast the space of these *natural* images, they occupy a tiny, minuscule fraction of the total image space. The majority of the images in image space represent random noise, resembling television static. For simplicity, consider grayscale images, and imagine drawing a random number for the gray value of each of the pixels. With exceedingly high probability, the resulting image will look like noise, with no apparent significance. You could draw these random images for an entire lifetime and never find an image of a mountain, or a person, or anything physically recognizable.

In other words, natural images are extremely rare in the vastness of image space. Because so many images are unstructured or random, most of the dimensions used to encode images are only necessary for these random images. These dimensions would be redundant if all we cared about was encoding natural images. An important implication is that the images we care about (i.e., natural images) are highly compressible, as long as we find a suitable transformed based where the redundant dimensions are easily identified. These ideas of compressibility will be explore in much greater detail in the context of Fourier Transforms in Chapter **??** and Compressive Sensing in Chapter **??**.

1.3 Mathematical Properties and Manipulations

Here we describe important mathematical properties of the SVD including geometric interpretations of the unitary matrices $\mathbf{U}$ and $\mathbf{V}$ as well as a discussion of the SVD in terms of dominant correlations in the data $\mathbf{X}$. The relationship between the SVD and correlations in the data will be explored more in Sec. 1.5 on Principal Components Analysis.

1.3.1 Interpretation as Dominant Correlations

The SVD is closely related to an eigenvalue problem involving the correlation matrices $\mathbf{XX}^*$ and $\mathbf{X}^*\mathbf{X}$, shown in Fig. 1.5. If we plug in Eq. 1.3 to the row-wise correlation matrix $\mathbf{XX}^*$ and the column-wise correlation matrix $\mathbf{X}^*\mathbf{X}$, we find:

$$\mathbf{XX}^* = \mathbf{U}\begin{bmatrix}\hat{\mathbf{\Sigma}}\\ \mathbf{0}\end{bmatrix}\mathbf{V}^*\mathbf{V}\begin{bmatrix}\hat{\mathbf{\Sigma}} & \mathbf{0}\end{bmatrix}\mathbf{U}^* = \mathbf{U}\begin{bmatrix}\hat{\mathbf{\Sigma}}^2 & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{U}^* \tag{1.7a}$$

$$\mathbf{X}^*\mathbf{X} = \mathbf{V}\begin{bmatrix}\hat{\mathbf{\Sigma}} & \mathbf{0}\end{bmatrix}\mathbf{U}^*\mathbf{U}\begin{bmatrix}\hat{\mathbf{\Sigma}}\\ \mathbf{0}\end{bmatrix}\mathbf{V}^* = \mathbf{V}\hat{\mathbf{\Sigma}}^2\mathbf{V}^*. \tag{1.7b}$$

Therefore, the matrices $\mathbf{U}, \mathbf{\Sigma}$, and $\mathbf{V}$ are solutions to the following eigenvalue problems:

$$\mathbf{XX}^*\mathbf{U} = \mathbf{U}\begin{bmatrix}\hat{\mathbf{\Sigma}}^2 & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}, \tag{1.8a}$$

$$\mathbf{X}^*\mathbf{XV} = \mathbf{V}\hat{\mathbf{\Sigma}}^2. \tag{1.8b}$$

This provides an intuitive interpretation of the SVD, where the columns of $\mathbf{U}$ are eigenvectors of the correlation matrix $\mathbf{XX}^*$ and columns of $\mathbf{V}$ are eigenvectors of $\mathbf{X}^*\mathbf{X}$. Therefore, columns of $\mathbf{U}$ are ordered in terms of the vectors that most describe correlation among columns of $\mathbf{X}$, and likewise with $\mathbf{V}$ and rows of $\mathbf{X}$.

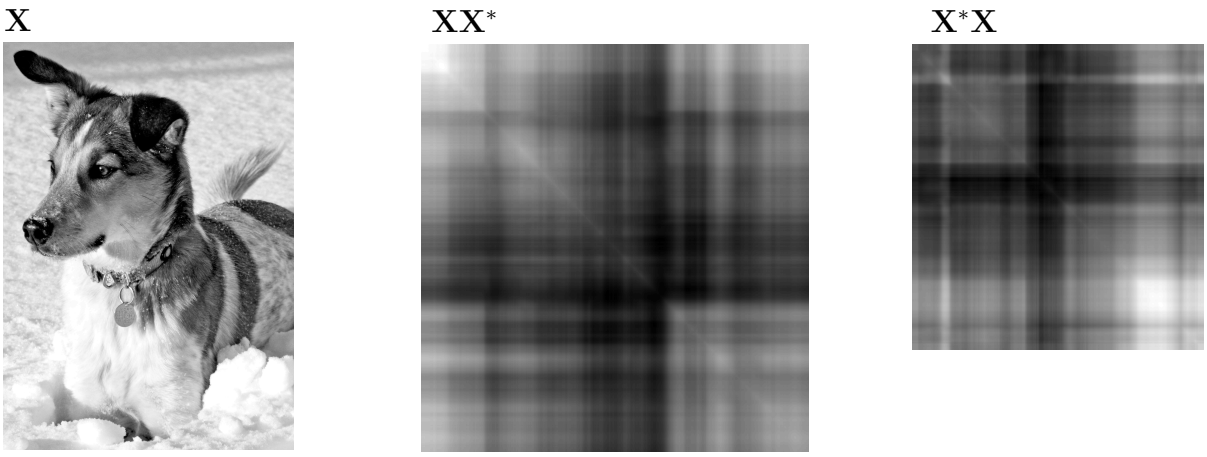

Figure 1.5: Correlation matrices $\mathbf{XX}^*$ and $\mathbf{X}^*\mathbf{X}$ for a matrix $\mathbf{X}$ obtained from an image of Mordecai the snow dog. Note that both correlation matrices are symmetric.