

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/332751929>

Chapter 1: Singular Value Decomposition

Chapter · April 2019

CITATIONS

2

READS

7,729

2 authors, including:



Steven L. Brunton

University of Washington Seattle

294 PUBLICATIONS 11,537 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Chaos as an intermittently forced linear system [View project](#)



machine learning: control, prediction of dynamical systems [View project](#)

Chapter 1: Singular Value Decomposition

from

Data Driven Science & Engineering

Machine Learning, Dynamical Systems, and Control

Cambridge University Press, 2019

Steven L. Brunton
Mechanical Engineering
University of Washington

J. Nathan Kutz
Applied Mathematics
University of Washington

Book website: databookuw.com

Contents

Preface	vi
Common Optimization Techniques, Equations, Symbols, and Acronyms	x
I Dimensionality Reduction and Transforms	1
1 Singular Value Decomposition	3
1.1 Overview	4
1.2 Matrix approximation	8
1.3 Mathematical properties and manipulations	12
1.4 Pseudo-inverse, least-squares, and regression	17
1.5 Principal component analysis (PCA)	24
1.6 Eigenfaces example	29
1.7 Truncation and alignment	34
1.8 Randomized singular value decomposition	42
1.9 Tensor decompositions and N -way data arrays	47
2 Fourier and Wavelet Transforms	54
2.1 Fourier series and Fourier transforms	55
2.2 Discrete Fourier transform (DFT) and fast Fourier transform (FFT)	65
2.3 Transforming partial differential equations	73
2.4 Gabor transform and the spectrogram	80
2.5 Wavelets and multi-resolution analysis	85
2.6 2D transforms and image processing	88
3 Sparsity and Compressed Sensing	96
3.1 Sparsity and compression	97
3.2 Compressed sensing	101
3.3 Compressed sensing examples	106
3.4 The geometry of compression	110
3.5 Sparse regression	113
3.6 Sparse representation	118
3.7 Robust principal component analysis (RPCA)	123

3.8 Sparse sensor placement	125
II Machine Learning and Data Analysis	132
4 Regression and Model Selection	134
4.1 Classic curve fitting	136
4.2 Nonlinear regression and gradient descent	142
4.3 Regression and $\mathbf{Ax} = \mathbf{b}$: Over- and under-determined systems . .	149
4.4 Optimization as the cornerstone of regression	155
4.5 The Pareto front and <i>Lex Parsimoniae</i>	162
4.6 Model selection: Cross validation	166
4.7 Model selection: Information criteria	172
5 Clustering and Classification	178
5.1 Feature selection and data mining	179
5.2 Supervised versus unsupervised learning	185
5.3 Unsupervised learning: k -means clustering	189
5.4 Unsupervised hierarchical clustering: Dendrogram	194
5.5 Mixture models and the expectation-maximization algorithm . .	198
5.6 Supervised learning and linear discriminants	203
5.7 Support vector machines (SVM)	209
5.8 Classification trees and random forest	214
5.9 Top 10 algorithms in data mining 2008	220
6 Neural Networks and Deep Learning	226
6.1 Neural networks: 1-Layer networks	227
6.2 Multi-layer networks and activation functions	232
6.3 The backpropagation algorithm	237
6.4 The stochastic gradient descent algorithm	242
6.5 Deep convolutional neural networks	245
6.6 Neural networks for dynamical systems	250
6.7 The diversity of neural networks	255
III Dynamics and Control	264
7 Data-Driven Dynamical Systems	266
7.1 Overview, motivations, and challenges	267
7.2 Dynamic mode decomposition (DMD)	274
7.3 Sparse identification of nonlinear dynamics (SINDy)	288
7.4 Koopman operator theory	299
7.5 Data-driven Koopman analysis	312

8 Linear Control Theory	323
8.1 Closed-loop feedback control	325
8.2 Linear time-invariant systems	330
8.3 Controllability and observability	336
8.4 Optimal full-state control: linear quadratic regulator (LQR)	343
8.5 Optimal full-state estimation: The Kalman filter	347
8.6 Optimal sensor-based control: Linear quadratic Gaussian (LQG)	350
8.7 Case study: Inverted pendulum on a cart	352
8.8 Robust control and frequency domain techniques	362
9 Balanced Models for Control	376
9.1 Model reduction and system identification	376
9.2 Balanced model reduction	377
9.3 System identification	393
10 Data-Driven Control	405
10.1 Nonlinear system identification for control	406
10.2 Machine learning control	414
10.3 Adaptive extremum-seeking control	427
IV Reduced Order Models	438
11 Reduced Order Models (ROMs)	440
11.1 POD for partial differential equations	440
11.2 Optimal basis elements: The POD expansion	447
11.3 POD and soliton dynamics	453
11.4 Continuous formulation of POD	459
11.5 POD with symmetries: Rotations and translations	464
12 Interpolation for Parametric ROMs	473
12.1 Gappy POD	473
12.2 Error and convergence of gappy POD	480
12.3 Gappy measurements: Minimize condition number	485
12.4 Gappy measurements: Maximal variance	492
12.5 POD and the discrete empirical interpolation method (DEIM)	497
12.6 DEIM algorithm implementation	501
12.7 Machine learning ROMs	504
Glossary	512
References	521

Chapter 1

Singular Value Decomposition (SVD)

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 in this book. The SVD provides a numerically stable matrix decomposition that can be used for a variety of purposes and is guaranteed to exist. We will use the SVD to obtain low-rank approximations to matrices and to perform pseudo-inverses of non-square matrices to find the solution of a system of equations $\mathbf{Ax} = \mathbf{b}$. Another important use of the SVD is as the underlying algorithm of principal component analysis (PCA), where high-dimensional data is decomposed into its most statistically descriptive factors. SVD/PCA has been applied to a wide variety of problems in science and engineering.

In a sense, the SVD generalizes the concept of the fast Fourier transform (FFT), which will be the subject of the next chapter. Many engineering texts begin with the FFT, as it is the basis of many classical analytical and numerical results. However, the FFT works in idealized settings, and the SVD is a more generic data-driven technique. Because this book is focused on data, we begin with the SVD, which may be thought of as providing a basis that is *tailored* to the specific data, as opposed to the FFT, which provides a *generic* basis.

In many domains, complex systems will generate data that is naturally arranged in large matrices, or more generally in arrays. For example, a time-series of data from an experiment or a simulation may be arranged in a matrix with each column containing all of the measurements at a given time. If the data at each instant in time is multi-dimensional, as in a high-resolution simulation of the weather in three spatial dimensions, it is possible to reshape or *flatten* this data into a high-dimensional column vector, forming the columns of a large matrix. Similarly, the pixel values in a grayscale image may be stored in a matrix, or these images may be reshaped into large column vectors in a matrix to represent the frames of a movie. Remarkably, the data generated by these systems are typically low rank, meaning that there are a few dominant patterns that explain the high-dimensional data. The SVD is a numerically robust and efficient method of extracting these patterns from data.

1.1 Overview

Here we introduce the singular value decomposition (SVD) and develop an intuition for how to apply the SVD by demonstrating its use on a number of motivating examples. The SVD will provide a foundation for many other techniques developed in this book, including classification methods in Chapter 5, the dynamic mode decomposition (DMD) in Chapter 7, and the proper orthogonal decomposition (POD) in Chapter 11. Detailed mathematical properties are discussed in the following sections.

High dimensionality is a common challenge in processing data from complex systems. These systems may involve large measured data sets including audio, image, or video data. The data may also be generated from a physical system, such as neural recordings from a brain, or fluid velocity measurements from a simulation or experiment. In many naturally occurring systems, it is observed that data exhibit dominant patterns, which may be characterized by a low-dimensional attractor or manifold [251, 252].

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 Earth's atmosphere or the turbulent wake behind a vehicle also provide compelling examples of 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 a low-dimensional approximation to high-dimensional data in terms of dominant patterns. This technique is *data-driven* in that patterns are discovered purely from data, without the addition of expert knowledge or intuition. The SVD is numerically stable and provides a hierarchical representation of the data in terms of a new coordinate system defined by dominant correlations within the data. Moreover, the SVD is guaranteed to exist for any matrix, unlike the eigendecomposition.

The SVD has many powerful applications beyond dimensionality reduction of high-dimensional data. It is used to compute the pseudo-inverse of non-square matrices, providing solutions to underdetermined or overdetermined matrix equations, $\mathbf{Ax} = \mathbf{b}$. We will also use the SVD to de-noise data sets. The SVD is likewise 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.

Definition of the SVD

Generally, we are interested in analyzing a large data set $\mathbf{X} \in \mathbb{C}^{n \times m}$:

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_m \\ | & | & & | \end{bmatrix}. \quad (1.1)$$

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 a set of discrete points, a set of neural measurements, or the state of a weather simulation with one square kilometer resolution.

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 of degrees of freedom. 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 $\mathbf{X} \in \mathbb{C}^{n \times m}$:

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^* \quad (1.2)$$

where $\mathbf{U} \in \mathbb{C}^{n \times n}$ and $\mathbf{V} \in \mathbb{C}^{m \times m}$ are *unitary* matrices¹ with orthonormal columns, and $\Sigma \in \mathbb{R}^{n \times m}$ is a matrix with real, non-negative entries on the diagonal and zeros off the diagonal. Here $*$ denotes the complex conjugate transpose². As we will discover throughout this chapter, the condition that \mathbf{U} and \mathbf{V} are unitary is used extensively.

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

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

The full SVD and economy SVD are shown in Fig. 1.1. The columns of $\hat{\mathbf{U}}^\perp$ span a vector space that is complementary and orthogonal to that spanned by $\hat{\mathbf{U}}$. 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 $\hat{\Sigma} \in \mathbb{C}^{m \times m}$ are called *singular values* and they are ordered from largest to smallest. The rank of \mathbf{X} is equal to the number of non-zero singular values.

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

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

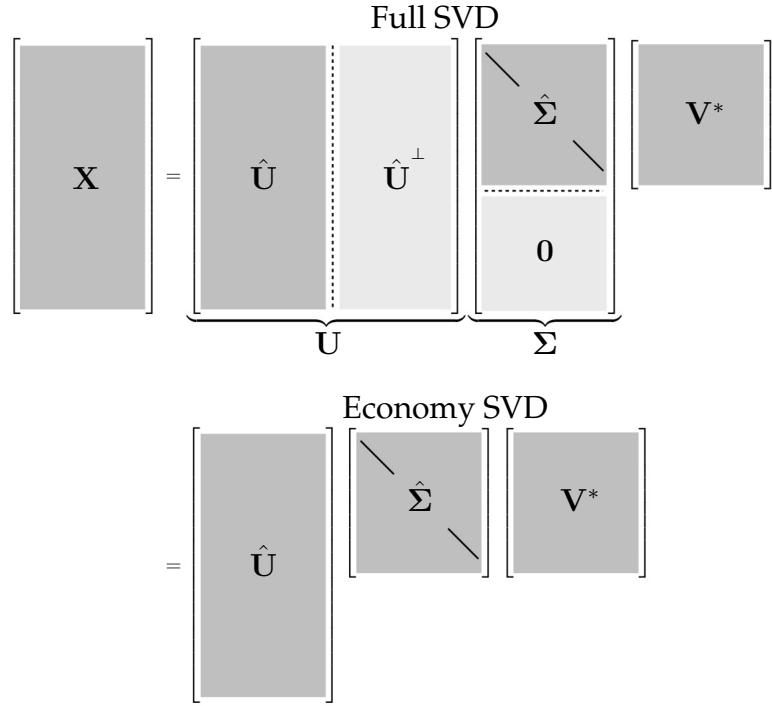


Figure 1.1: Schematic of matrices in the full and economy SVD.

Computing the SVD

The SVD is a cornerstone of computational science and engineering, and the numerical implementation of the SVD is both important and mathematically enlightening. That said, most standard numerical implementations are mature and a simple interface exists in many modern computer languages, allowing us to abstract away the details underlying the SVD computation. For most purposes, we simply use the SVD as a part of a larger effort, and we take for granted the existence of efficient and stable numerical algorithms. In the sections below, we demonstrate how to use the SVD in various computational languages, and we also discuss the most common computational strategies and limitations. There are numerous important results on the computation of the SVD [211, 106, 213, 292, 238]. A more thorough discussion of computational issues can be found in [214]. Randomized numerical algorithms are increasingly used to compute the SVD of very large matrices as discussed in Sec. 1.8.

Matlab. In Matlab, computing the SVD is straightforward:

```
>>X = randn(5,3); % Create a 5x3 random data matrix
>>[U,S,V] = svd(X); % Singular Value Decomposition
```

For non-square matrices \mathbf{X} , the economy SVD is more efficient:

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

Python

```
>>> import numpy as np
>>> X = np.random.rand(5, 3) % create random data matrix
>>> U, S, V = np.linalg.svd(X, full_matrices=True) % full SVD
>>> Uhat, Shat, Vhat = np.linalg.svd(X, full_matrices=False)
      % economy SVD
```

R

```
> X <- replicate(3, rnorm(5))
> s <- svd(X)
> U <- s$u
> S <- diag(s$d)
> V <- s$v
```

Mathematica

```
In:= X=RandomReal[{0,1},{5,3}]
In:= {U,S,V} = SingularValueDecomposition[X]
```

Other languages

The SVD is also available in other languages, such as Fortran and C++. In fact, most SVD implementations are based on the LAPACK (Linear Algebra Package) [13] in Fortran. The SVD routine is designated **DGESVD** in LAPACK, and this is wrapped in the C++ libraries **Armadillo** and **Eigen**.

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 [502], 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 seminal computational work of Golub and collaborators [211, 213]. In addition, there are many excellent chapters on the SVD in modern texts [524, 17, 316].

Uses in this book and assumptions of the reader

The SVD is the basis for many related techniques in dimensionality reduction. These methods include principal component analysis (PCA) in statistics [418, 256, 257], the Karhunen–Loëve transform (KLT) [280, 340], empirical orthogonal functions (EOFs) in climate [344], the proper orthogonal de-

composition (POD) in fluid dynamics [252], and canonical correlation analysis (CCA) [131]. 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 by Gerbrands [204].

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 [388].

For this chapter, we assume that the reader is familiar with linear algebra with some experience in computation and numerics. For review, there are a number of excellent books on numerical linear algebra, with discussions on the SVD [524, 17, 316].

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} [476]. Schmidt's approximation theorem was rediscovered by Eckart and Young [170], and is sometimes referred to as the Eckart-Young theorem.

Theorem 1 (Eckart-Young [170]) *The optimal rank- r approximation to \mathbf{X} , in a least-squares sense, is given by the rank- r SVD truncation $\tilde{\mathbf{X}}$:*

$$\underset{\tilde{\mathbf{X}}, \text{s.t. } \text{rank}(\tilde{\mathbf{X}})=r}{\operatorname{argmin}} \|\mathbf{X} - \tilde{\mathbf{X}}\|_F = \tilde{\mathbf{U}} \tilde{\Sigma} \tilde{\mathbf{V}}^*. \quad (1.4)$$

Here, $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ denote the first r leading columns of \mathbf{U} and \mathbf{V} , and $\tilde{\Sigma}$ contains the leading $r \times r$ sub-block of Σ . $\|\cdot\|_F$ is the Frobenius norm.

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

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

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

$\tilde{\mathbf{X}}$. Thus, high-dimensional data may be well described by a few dominant patterns given by the columns of $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$.

This is an 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 $\tilde{\mathbf{U}}$ 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* (see Chapter 7), and the SVD basis provides a hierarchy of modes that characterize the observed attractor, on which we may project a low-dimensional dynamical system to obtain reduced order models (see Chapter 12).

Truncation

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

$$\mathbf{X} \approx \tilde{\mathbf{U}}\tilde{\Sigma}\tilde{\mathbf{V}}^*. \quad (1.6)$$

There are numerous choices for the truncation rank r , and they are discussed in Sec. 1.7. If we choose the truncation value to keep all non-zero singular values, then $\mathbf{X} = \tilde{\mathbf{U}}\tilde{\Sigma}\tilde{\mathbf{V}}^*$ is exact.

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 directions, respectively³. Depending on the basis of representation (pixel-space, Fourier frequency domain, SVD transform coordinates), images may have very compact approximations.

³It 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.

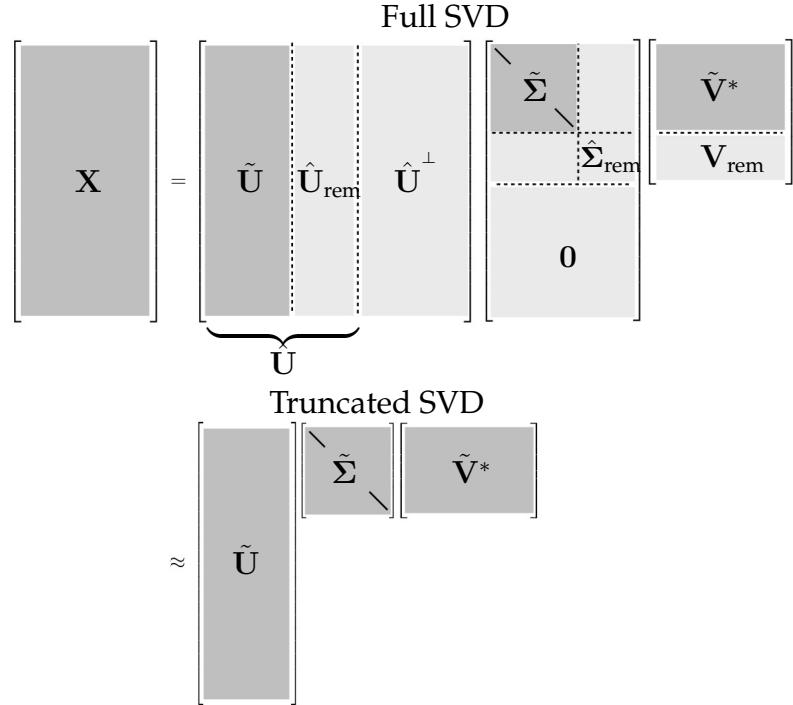


Figure 1.2: Schematic of truncated SVD. The subscript '_{rem}' denotes the remainder of $\hat{\mathbf{U}}$, $\hat{\Sigma}$ or \mathbf{V} after truncation.

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 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 Σ , must be stored in $\tilde{\mathbf{U}}$, $\tilde{\Sigma}$ and $\tilde{\mathbf{V}}$.

First, we load the image:

```
A=imread('..../DATA/dog.jpg');
X=double(rgb2gray(A)); % Convert RGB->gray, 256 bit->double.
nx = size(X,1); ny = size(X,2);
imagesc(X), axis off, colormap gray
```

and take the SVD:

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

Next, we compute the approximate matrix using the truncated SVD for various ranks ($r = 5, 20$, and 100):

```
for r=[5 20 100]; % Truncation value
    Xapprox = U(:,1:r)*S(1:r,1:r)*V(:,1:r)'; % Approx. image
```

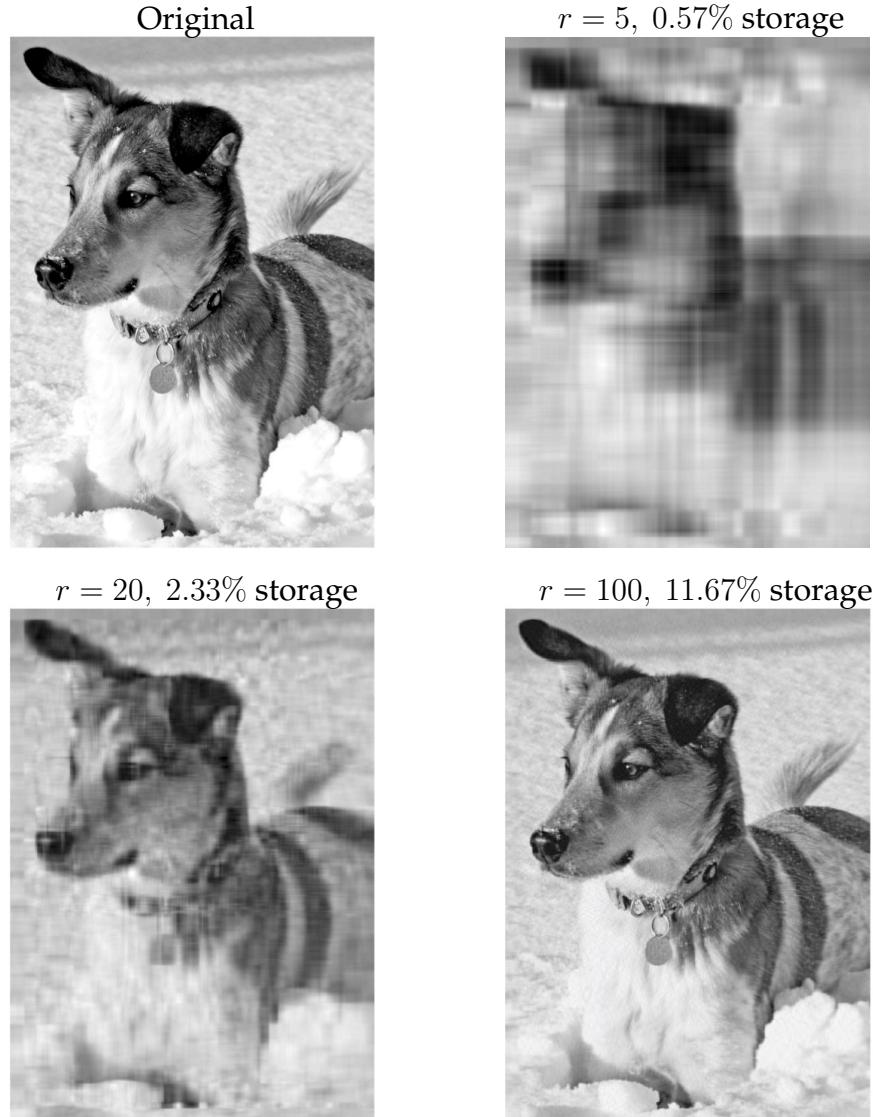


Figure 1.3: Image compression of Mordecai the snow dog, truncating the SVD at various ranks r . Original image resolution is 2000×1500 .

```

|| figure, imagesc(Xapprox), axis off
|| title(['r=',num2str(r,'%d'),']);
```

Finally, we plot the singular values and cumulative energy in Fig. 1.4:

```

|| subplot(1,2,1), semilogy(diag(S),'k')
|| subplot(1,2,2), plot(cumsum(diag(S))/sum(diag(S)),'k')
```

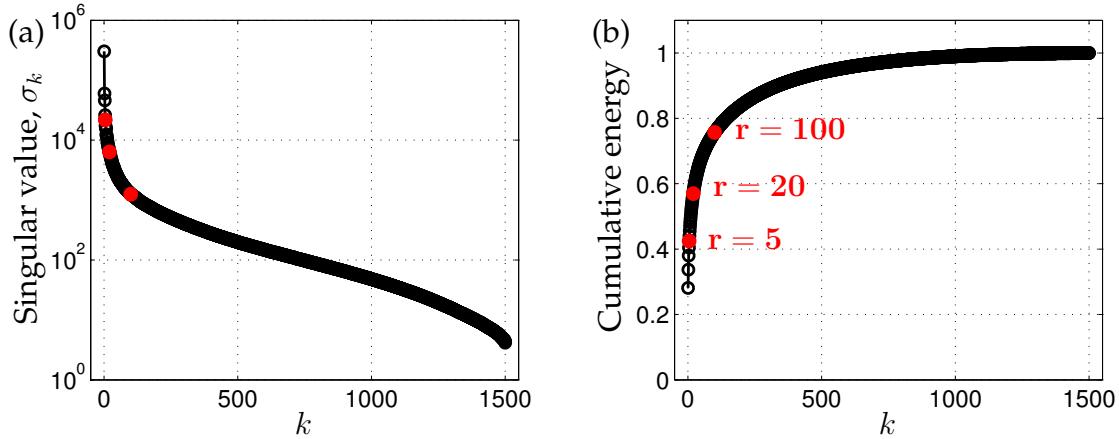


Figure 1.4: (a) Singular values σ_k . (b) Cumulative energy in the first k modes.

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.

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 for a specific image, and in Figs. 1.6 and 1.7 for generic matrices. If we plug (1.3) into the row-wise correlation matrix \mathbf{XX}^* and the column-wise correlation matrix $\mathbf{X}^*\mathbf{X}$, we find:

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

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

Recalling that \mathbf{U} and \mathbf{V} are unitary, \mathbf{U} , Σ , and \mathbf{V} are solutions to the following eigenvalue problems:

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

$$\mathbf{X}^* \mathbf{X} \mathbf{V} = \mathbf{V} \hat{\Sigma}^2. \quad (1.8b)$$

In other words, each nonzero singular value of \mathbf{X} is a positive square root of an eigenvalue of $\mathbf{X}^*\mathbf{X}$ and of \mathbf{XX}^* , which have the same non-zero eigenvalues. It

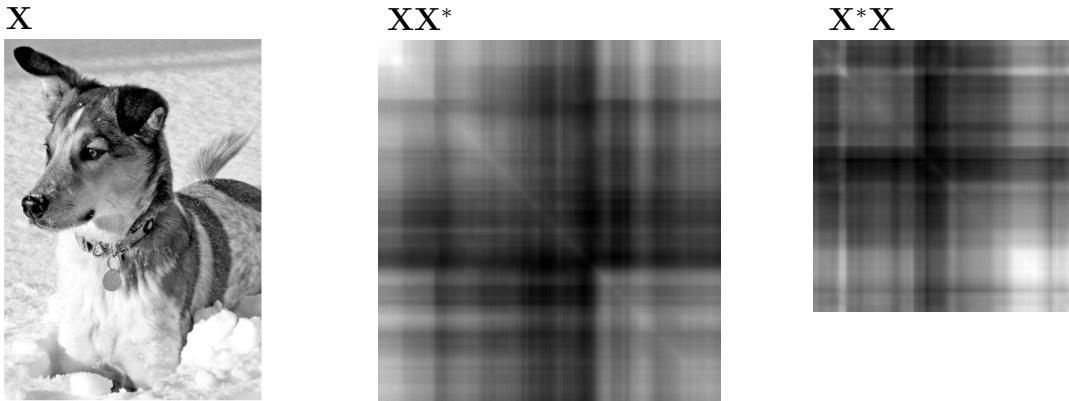


Figure 1.5: Correlation matrices $\mathbf{X}\mathbf{X}^*$ and $\mathbf{X}^*\mathbf{X}$ for a matrix \mathbf{X} obtained from an image of a dog. Note that both correlation matrices are symmetric.

follows that if \mathbf{X} is self-adjoint (i.e. $\mathbf{X} = \mathbf{X}^*$), then the singular values of \mathbf{X} are equal to the absolute value of the eigenvalues of \mathbf{X} .

This provides an intuitive interpretation of the SVD, where the columns of \mathbf{U} are eigenvectors of the correlation matrix $\mathbf{X}\mathbf{X}^*$ and columns of \mathbf{V} are eigenvectors of $\mathbf{X}^*\mathbf{X}$. We choose to arrange the singular values in descending order by magnitude, and thus the columns of \mathbf{U} are hierarchically ordered by how much correlation they capture in the columns of \mathbf{X} ; \mathbf{V} similarly captures correlation in the rows of \mathbf{X} .

Method of snapshots

It is often impractical to construct the matrix $\mathbf{X}\mathbf{X}^*$ because of the large size of the state-dimension n , let alone solve the eigenvalue problem; if \mathbf{x} has a million elements, then $\mathbf{X}\mathbf{X}^*$ has a trillion elements. In 1987, Sirovich observed that it is possible to bypass this large matrix and compute the first m columns of \mathbf{U} using what is now known as the method of snapshots [490].

Instead of computing the eigen-decomposition of $\mathbf{X}\mathbf{X}^*$ to obtain the left singular vectors \mathbf{U} , we only compute the eigen-decomposition of $\mathbf{X}^*\mathbf{X}$, which is much smaller and more manageable. From (1.8b), we then obtain \mathbf{V} and $\hat{\Sigma}$. If there are zero singular values in $\hat{\Sigma}$, then we only keep the r non-zero part, $\tilde{\Sigma}$, and the corresponding columns $\tilde{\mathbf{V}}$ of \mathbf{V} . From these matrices, it is then possible to approximate $\tilde{\mathbf{U}}$, the first r columns of \mathbf{U} , as follows:

$$\tilde{\mathbf{U}} = \mathbf{X}\tilde{\mathbf{V}}\tilde{\Sigma}^{-1}. \quad (1.9)$$

$$\begin{bmatrix} \mathbf{X} \\ \vdots \\ \mathbf{X}^* \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{X}^* \\ \mathbf{X} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{XX}^* \\ \vdots \\ \mathbf{X}^*\mathbf{X} \end{bmatrix}$$

Figure 1.6: Correlation matrix \mathbf{XX}^* is formed by taking the inner product of rows of \mathbf{X} .

$$\begin{bmatrix} \mathbf{X}^* \\ \vdots \\ \mathbf{X} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{X}^* \\ \mathbf{X} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{X}^*\mathbf{X} \\ \vdots \\ \mathbf{XX}^* \end{bmatrix}$$

Figure 1.7: Correlation matrix $\mathbf{X}^*\mathbf{X}$ is formed by taking the inner product of columns of \mathbf{X} .

Geometric interpretation

The columns of the matrix \mathbf{U} provide an orthonormal basis for the column space of \mathbf{X} . Similarly, the columns of \mathbf{V} provide an orthonormal basis for the row space of \mathbf{X} . If the columns of \mathbf{X} are spatial measurements in time, then \mathbf{U} encode spatial patterns, and \mathbf{V} encode temporal patterns.

One property that makes the SVD particularly useful is the fact that both \mathbf{U} and \mathbf{V} are *unitary* matrices, so that $\mathbf{UU}^* = \mathbf{U}^*\mathbf{U} = \mathbf{I}_{n \times n}$ and $\mathbf{VV}^* = \mathbf{V}^*\mathbf{V} = \mathbf{I}_{m \times m}$. This means that solving a system of equations involving \mathbf{U} or \mathbf{V} is as simple as multiplication by the transpose, which scales as $\mathcal{O}(n^2)$, as opposed to traditional methods for the generic inverse, which scale as $\mathcal{O}(n^3)$. As noted in the previous section and in [57], the SVD is intimately connected to the spectral properties of the compact self-adjoint operators \mathbf{XX}^* and $\mathbf{X}^*\mathbf{X}$.

The SVD of \mathbf{X} may be interpreted geometrically based on how a hyper-sphere, given by $S^{n-1} \triangleq \{\mathbf{x} \mid \|\mathbf{x}\|_2 = 1\} \subset \mathbb{R}^n$ maps into an ellipsoid, $\{\mathbf{y} \mid \mathbf{y} = \mathbf{X}\mathbf{x} \text{ for } \mathbf{x} \in S^{n-1}\} \subset \mathbb{R}^m$, through \mathbf{X} . This is shown graphically in Fig. 1.8 for a sphere in \mathbb{R}^3 and a mapping \mathbf{X} with three non-zero singular values. Because the mapping through \mathbf{X} (i.e., matrix multiplication) is linear, knowing how it maps the unit sphere determines how all other vectors will map.

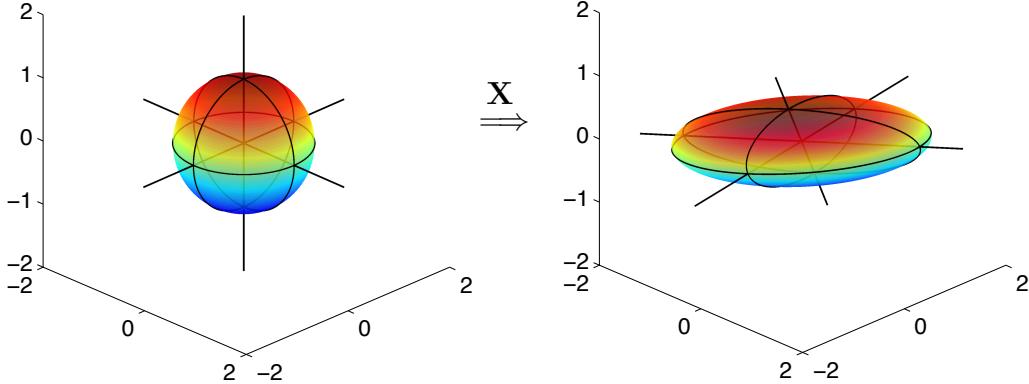


Figure 1.8: Geometric illustration of the SVD as a mapping from a sphere in \mathbb{R}^n to an ellipsoid in \mathbb{R}^m .

For the specific case shown in Fig. 1.8, we construct the matrix X out of three rotation matrices, R_x , R_y , and R_z , and a fourth matrix to stretch out and scale the principal axes:

$$X = \underbrace{\begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{R_z} \underbrace{\begin{bmatrix} \cos(\theta_2) & 0 & \sin(\theta_2) \\ 0 & 1 & 0 \\ -\sin(\theta_2) & 0 & \cos(\theta_2) \end{bmatrix}}_{R_y} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_1) & -\sin(\theta_1) \\ 0 & \sin(\theta_1) & \cos(\theta_1) \end{bmatrix}}_{R_x} \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}.$$

In this case, $\theta_1 = \pi/15$, $\theta_2 = -\pi/9$, and $\theta_3 = -\pi/20$, and $\sigma_1 = 3$, $\sigma_2 = 1$, and $\sigma_3 = 0.5$. These rotation matrices do not commute, and so the order of rotation matters. If one of the singular values is zero, then a dimension is removed and the ellipsoid collapses onto a lower-dimensional subspace. The product $R_x R_y R_z$ is the unitary matrix U in the SVD of X . The matrix V is the identity.

Code 1.1: Construct rotation matrices.

```

theta = [pi/15; -pi/9; -pi/20];
Sigma = diag([3; 1; 0.5]); % scale x, y, and z

Rx = [1 0 0; % rotate about x-axis
      0 cos(theta(1)) -sin(theta(1));
      0 sin(theta(1)) cos(theta(1))];

Ry = [cos(theta(2)) 0 sin(theta(2)); % rotate about y-axis
      0 1 0;
      -sin(theta(2)) 0 cos(theta(2))];

Rz = [cos(theta(3)) -sin(theta(3)) 0; % rotate about z-axis
      sin(theta(3)) cos(theta(3)) 0;
      0 0 1];

X = Rz*Ry*Rx*Sigma; % rotate and scale

```

Code 1.2: Plot sphere.

```
[x, y, z] = sphere(25);
h1=surf(x, y, z);
```

Code 1.3: Map sphere through X and plot resulting ellipsoid.

```
xR = 0*x; yR = 0*y; zR = 0*z;
for i=1:size(x,1)
    for j=1:size(x,2)
        vecR = X*[x(i,j); y(i,j); z(i,j)];
        xR(i,j) = vecR(1);
        yR(i,j) = vecR(2);
        zR(i,j) = vecR(3);
    end
end
h2=surf(xR, yR, zR, z); % using sphere z-coord for color
```

Invariance of the SVD to unitary transformations

A useful property of the SVD is that if we left or right-multiply our data matrix \mathbf{X} by a unitary transformation, it preserves the terms in the SVD, except for the corresponding left or right unitary matrix \mathbf{U} or \mathbf{V} , respectively. This has important implications, since the discrete Fourier transform (DFT; see Chapter 2) \mathcal{F} is a unitary transform, meaning that the SVD of data $\hat{\mathbf{X}} = \mathcal{F}\mathbf{X}$ will be exactly the same as the SVD of \mathbf{X} , except that the modes $\hat{\mathbf{U}}$ will be the DFT of modes \mathbf{U} : $\hat{\mathbf{U}} = \mathcal{F}\mathbf{U}$. In addition, the invariance of the SVD to unitary transformations enable the use of compressed measurements to reconstruct SVD modes that are sparse in some transform basis (see Chapter 3).

The invariance of SVD to unitary transformations is geometrically intuitive, as unitary transformations rotate vectors in space, but do not change their inner products or correlation structures. We denote a left unitary transformation by \mathbf{C} , so that $\mathbf{Y} = \mathbf{C}\mathbf{X}$, and a right unitary transformation by \mathbf{P}^* , so that $\mathbf{Y} = \mathbf{X}\mathbf{P}^*$. The SVD of \mathbf{X} will be denoted $\mathbf{U}_\mathbf{X}\Sigma_\mathbf{X}\mathbf{V}_\mathbf{X}^*$ and the SVD of \mathbf{Y} will be $\mathbf{U}_\mathbf{Y}\Sigma_\mathbf{Y}\mathbf{V}_\mathbf{Y}^*$.

Left unitary transformations

First, consider a left unitary transformation of \mathbf{X} : $\mathbf{Y} = \mathbf{C}\mathbf{X}$. Computing the correlation matrix $\mathbf{Y}^*\mathbf{Y}$, we find

$$\mathbf{Y}^*\mathbf{Y} = \mathbf{X}^*\mathbf{C}^*\mathbf{C}\mathbf{X} = \mathbf{X}^*\mathbf{X}. \quad (1.10)$$

The projected data has the same eigendecomposition, resulting in the same $\mathbf{V}_\mathbf{X}$ and $\Sigma_\mathbf{X}$. Using the method of snapshots to reconstruct $\mathbf{U}_\mathbf{Y}$, we find

$$\mathbf{U}_\mathbf{Y} = \mathbf{Y}\mathbf{V}_\mathbf{X}\Sigma_\mathbf{X}^{-1} = \mathbf{C}\mathbf{X}\mathbf{V}_\mathbf{X}\Sigma_\mathbf{X}^{-1} = \mathbf{C}\mathbf{U}_\mathbf{X}. \quad (1.11)$$

Thus, $\mathbf{U}_Y = \mathbf{C}\mathbf{U}_X$, $\Sigma_Y = \Sigma_X$, and $\mathbf{V}_Y = \mathbf{V}_X$. The SVD of \mathbf{Y} is then:

$$\mathbf{Y} = \mathbf{C}\mathbf{X} = \mathbf{C}\mathbf{U}_X \Sigma_X \mathbf{V}_X^*. \quad (1.12)$$

Right unitary transformations

For a right unitary transformation $\mathbf{Y} = \mathbf{X}\mathbf{P}^*$, the correlation matrix $\mathbf{Y}^*\mathbf{Y}$ is:

$$\mathbf{Y}^*\mathbf{Y} = \mathbf{P}\mathbf{X}^*\mathbf{X}\mathbf{P}^* = \mathbf{P}\mathbf{V}_X \Sigma_X^2 \mathbf{V}_X^* \mathbf{P}^*, \quad (1.13)$$

with the following eigendecomposition

$$\mathbf{Y}^*\mathbf{Y}\mathbf{P}\mathbf{V}_X = \mathbf{P}\mathbf{V}_X \Sigma_X^2. \quad (1.14)$$

Thus, $\mathbf{V}_Y = \mathbf{P}\mathbf{V}_X$ and $\Sigma_Y = \Sigma_X$. We may use the method of snapshots to reconstruct \mathbf{U}_Y :

$$\mathbf{U}_Y = \mathbf{Y}\mathbf{P}\mathbf{V}_X \Sigma_X^{-1} = \mathbf{X}\mathbf{V}_X \Sigma_X^{-1} = \mathbf{U}_X. \quad (1.15)$$

Thus, $\mathbf{U}_Y = \mathbf{U}_X$, and we may write the SVD of \mathbf{Y} as:

$$\mathbf{Y} = \mathbf{X}\mathbf{P}^* = \mathbf{U}_X \Sigma_X \mathbf{V}_X^* \mathbf{P}^*. \quad (1.16)$$

1.4 Pseudo-inverse, least-squares, and regression

Many physical systems may be represented as a linear system of equations:

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad (1.17)$$

where the constraint matrix \mathbf{A} and vector \mathbf{b} are known, and the vector \mathbf{x} is unknown. If \mathbf{A} is a square, invertible matrix (i.e., \mathbf{A} has nonzero determinant), then there exists a unique solution \mathbf{x} for every \mathbf{b} . However, when \mathbf{A} is either singular or rectangular, there may be one, none, or infinitely many solutions, depending on the specific \mathbf{b} and the column and row spaces of \mathbf{A} .

First, consider the *underdetermined system*, where $\mathbf{A} \in \mathbb{C}^{n \times m}$ and $n \ll m$ (i.e., \mathbf{A} is a short-fat matrix), so that there are fewer equations than unknowns. This type of system is likely to have full column rank, since it has many more columns than are required for a linearly independent basis⁴. Generically, if a short-fat \mathbf{A} has full column rank, then there are infinitely many solutions \mathbf{x} for every \mathbf{b} . The system is called *underdetermined* because there are not enough values in \mathbf{b} to uniquely determine the higher-dimensional \mathbf{x} .

⁴It is easy to construct degenerate examples where a short-fat matrix does not have full column rank, such as $\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$.

Similarly, consider the *overdetermined system*, where $n \gg m$ (i.e., a tall-skinny matrix), so that there are more equations than unknowns. This matrix cannot have a full column rank, and so it is guaranteed that there are vectors \mathbf{b} that have no solution \mathbf{x} . In fact, there will only be a solution \mathbf{x} if \mathbf{b} is in the column space of \mathbf{A} , i.e. $\mathbf{b} \in \text{col}(\mathbf{A})$.

Technically, there may be some choices of \mathbf{b} that admit infinitely many solutions \mathbf{x} for a tall-skinny matrix \mathbf{A} and other choices of \mathbf{b} that admit zero solutions even for a short-fat matrix. The solution space to the system in (1.17) is determined by the four fundamental subspaces of $\mathbf{A} = \tilde{\mathbf{U}}\tilde{\Sigma}\tilde{\mathbf{V}}^*$, where the rank r is chosen to include all nonzero singular values:

- The column space, $\text{col}(\mathbf{A})$, is the span of the columns of \mathbf{A} , also known as the *range*. The column space of \mathbf{A} is the same as the column space of $\tilde{\mathbf{U}}$;
- The orthogonal complement to $\text{col}(\mathbf{A})$ is $\ker(\mathbf{A}^*)$, given by the column space of $\hat{\mathbf{U}}^\perp$ from Fig. 1.1;
- The row space, $\text{row}(\mathbf{A})$, is the span of the rows of \mathbf{A} , which is spanned by the columns of $\tilde{\mathbf{V}}$. The row space of \mathbf{A} is equal to $\text{row}(\mathbf{A}) = \text{col}(\mathbf{A}^*)$;
- The kernel space, $\ker(\mathbf{A})$, is the orthogonal complement to $\text{row}(\mathbf{A})$, and is also known as the *null space*. The null space is the subspace of vectors that map through \mathbf{A} to zero, i.e., $\mathbf{Ax} = \mathbf{0}$, given by $\text{col}(\hat{\mathbf{V}}^\perp)$.

More precisely, if $\mathbf{b} \in \text{col}(\mathbf{A})$ and if $\dim(\ker(\mathbf{A})) \neq 0$, then there are infinitely many solutions \mathbf{x} . Note that the condition $\dim(\ker(\mathbf{A})) \neq 0$ is guaranteed for a short-fat matrix. Similarly, if $\mathbf{b} \notin \text{col}(\mathbf{A})$, then there are no solutions, and the system of equations in (1.17) are called *inconsistent*.

The fundamental subspaces above satisfy the following properties:

$$\text{col}(\mathbf{A}) \oplus \ker(\mathbf{A}^*) = \mathbb{R}^n \quad (1.18a)$$

$$\text{col}(\mathbf{A}^*) \oplus \ker(\mathbf{A}) = \mathbb{R}^n. \quad (1.18b)$$

Remark 1 There is an extensive literature on random matrix theory, where the above stereotypes are almost certainly true, meaning that they are true with high probability. For example, a system $\mathbf{Ax} = \mathbf{b}$ is extremely unlikely to have a solution for a random matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ and random vector $\mathbf{b} \in \mathbb{R}^n$ with $n \gg m$, since there is little chance that \mathbf{b} is in the column space of \mathbf{A} . These properties of random matrices will play a prominent role in compressed sensing (see Chapter 3).

In the overdetermined case when no solution exists, we would often like to find the solution \mathbf{x} that minimizes the sum-squared error $\|\mathbf{Ax} - \mathbf{b}\|_2^2$, the so-called *least-squares* solution. Note that the least-squares solution also minimizes $\|\mathbf{Ax} - \mathbf{b}\|_2$. In the underdetermined case when infinitely many solutions exist,

we may like to find the solution \mathbf{x} with minimum norm $\|\mathbf{x}\|_2$ so that $\mathbf{Ax} = \mathbf{b}$, the so-called *minimum-norm* solution.

The SVD is the technique of choice for these important optimization problems. First, if we substitute an exact truncated SVD $\mathbf{A} = \tilde{\mathbf{U}}\tilde{\Sigma}\tilde{\mathbf{V}}^*$ in for \mathbf{A} , we can “invert” each of the matrices $\tilde{\mathbf{U}}$, $\tilde{\Sigma}$, and $\tilde{\mathbf{V}}^*$ in turn, resulting in the Moore-Penrose *left pseudo-inverse* [425, 426, 453, 572] \mathbf{A}^\dagger of \mathbf{A} :

$$\mathbf{A}^\dagger \triangleq \tilde{\mathbf{V}}\tilde{\Sigma}^{-1}\tilde{\mathbf{U}}^* \implies \mathbf{A}^\dagger\mathbf{A} = \mathbf{I}_{m \times m}. \quad (1.19)$$

This may be used to find both the minimum norm and least-squares solutions to (1.17):

$$\mathbf{A}^\dagger\mathbf{A}\tilde{\mathbf{x}} = \mathbf{A}^\dagger\mathbf{b} \implies \tilde{\mathbf{x}} = \tilde{\mathbf{V}}\tilde{\Sigma}^{-1}\tilde{\mathbf{U}}^*\mathbf{b}. \quad (1.20)$$

Plugging the solution $\tilde{\mathbf{x}}$ back in to (1.17) results in:

$$\mathbf{A}\tilde{\mathbf{x}} = \tilde{\mathbf{U}}\tilde{\Sigma}\tilde{\mathbf{V}}^*\tilde{\mathbf{V}}\tilde{\Sigma}^{-1}\tilde{\mathbf{U}}^*\mathbf{b} \quad (1.21a)$$

$$= \tilde{\mathbf{U}}\tilde{\mathbf{U}}^*\mathbf{b}. \quad (1.21b)$$

Note that $\tilde{\mathbf{U}}\tilde{\mathbf{U}}^*$ is not necessarily the identity matrix, but is rather a projection onto the column space of $\tilde{\mathbf{U}}$. Therefore, $\tilde{\mathbf{x}}$ will only be an exact solution to (1.17) when \mathbf{b} is in the column space of $\tilde{\mathbf{U}}$, and therefore in the column space of \mathbf{A} .

Computing the pseudo-inverse \mathbf{A}^\dagger is computationally efficient, after the expensive up-front cost of computing the SVD. Inverting the unitary matrices $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ involves matrix multiplication by the transpose matrices, which are $\mathcal{O}(n^2)$ operations. Inverting $\tilde{\Sigma}$ is even more efficient since it is a diagonal matrix, requiring $\mathcal{O}(n)$ operations. In contrast, inverting a dense square matrix would require an $\mathcal{O}(n^3)$ operation.

One-dimensional linear regression

Regression is an important statistical tool to relate variables to one another based on data [360]. Consider the collection of data in Fig. 1.9. The red ‘x’’s are obtained by adding Gaussian white noise to the black line, as shown in Code 1.4. We assume that the data is linearly related, as in (1.17), and we use the pseudo-inverse to find the least-squares solution for the slope x below (blue dashed line), shown in Code 1.5:

$$\begin{bmatrix} \mathbf{b} \\ \vdots \end{bmatrix} = \begin{bmatrix} \mathbf{a} \\ \vdots \end{bmatrix} x = \tilde{\mathbf{U}}\tilde{\Sigma}\tilde{\mathbf{V}}^*x. \quad (1.22a)$$

$$\implies x = \tilde{\mathbf{V}}\tilde{\Sigma}^{-1}\tilde{\mathbf{U}}^*\mathbf{b}. \quad (1.22b)$$

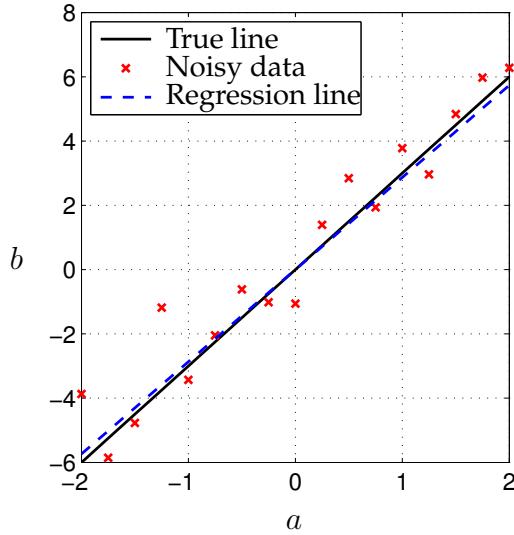


Figure 1.9: Illustration of linear regression using noisy data.

In (1.22b), $\tilde{\Sigma} = \|\mathbf{a}\|_2$, $\tilde{\mathbf{V}} = 1$, and $\tilde{\mathbf{U}} = \mathbf{a}/\|\mathbf{a}\|_2$. Taking the left pseudo-inverse:

$$x = \frac{\mathbf{a}^* \mathbf{b}}{\|\mathbf{a}\|_2^2}. \quad (1.23)$$

This makes physical sense, if we think of x as the value that best maps our vector \mathbf{a} to the vector \mathbf{b} . Then, the best single value x is obtained by taking the dot product of \mathbf{b} with the normalized \mathbf{a} direction. We then add a second normalization factor $\|\mathbf{a}\|_2$ because the \mathbf{a} in (1.22a) is not normalized.

Note that strange things happen if you use row vectors instead of column vectors in (1.22) above. Also, if the noise magnitude becomes large relative to the slope x , the pseudo-inverse will undergo a phase-change in accuracy, related to the hard-thresholding results in subsequent sections.

Code 1.4: Generate noisy data for Fig. 1.9.

```

x = 3; % True slope
a = [-2:.25:2]';
b = a*x + 1*randn(size(a)); % Add noise
plot(a,x*a,'k') % True relationship
hold on, plot(a,b,'rx') % Noisy measurements

```

Code 1.5: Compute least-squares approximation for Fig. 1.9.

```

[U,S,V] = svd(a,'econ');
xtilde = V*inv(S)*U'*b; % Least-square fit
plot(a,xtilde*a,'b--') % Plot fit

```

The procedure above is called *linear regression* in statistics. There is a **regress** command in Matlab, as well as a **pinv** command that may also be used.

Code 1.6: Alternative formulations of least-squares in Matlab.

```
xtilde1 = V*inv(S)*U'*b
xtilde2 = pinv(a)*b
xtilde3 = regress(b,a)
```

Multilinear regression

Example 1: Cement heat generation data

First, we begin with a simple built-in Matlab dataset that describes the heat generation for various cement mixtures comprised of four basic ingredients. In this problem, we are solving (1.17) where $A \in \mathbb{R}^{13 \times 4}$, since there are four ingredients and heat measurements for 13 unique mixtures. The goal is to determine the weighting x that relates the proportions of the four ingredients to the heat generation. It is possible to find the minimum error solution using the SVD, as shown in Code 1.7. Alternatives, using **regress** and **pinv**, are also explored.

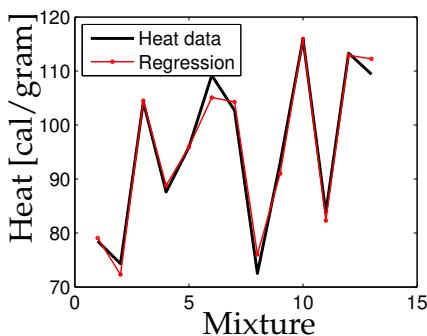
Code 1.7: Multilinear regression for cement heat data.

```
load hald; % Load Portland Cement dataset
A = ingredients;
b = heat;

[U,S,V] = svd(A,'econ');
x = V*inv(S)*U'*b; % Solve Ax=b using the SVD

plot(b,'k'); hold on % Plot data
plot(A*x,'r-o',); % Plot fit

x = regress(b,A); % Alternative 1 (regress)
x = pinv(A)*b; % Alternative 2 (pinv)
```



	Ingredient	Regression
1	Tricalcium aluminate	2.1930
2	Tricalcium silicate	1.1533
3	Tetracalcium alumiferrite	0.7585
4	Beta-dicalcium silicate	0.4863

Figure 1.10: Heat data for cement mixtures containing four basic ingredients.

Example 2: Boston housing data

In this example, we explore a larger data set to determine which factors best predict prices in the Boston housing market [234]. This data is available from the UCI Machine Learning Repository [24].

There are 13 attributes that are correlated with house price, such as per capita crime rate and property-tax rate. These features are regressed onto the price data, and the best fit price prediction is plotted against the true house value in Fig. 1.11, and the regression coefficients are shown in Fig. 1.12. Although the house value is not perfectly predicted, the trend agrees quite well. It is often the case that the highest value outliers are not well-captured by simple linear fits, as in this example.

This data contains prices and attributes for 506 homes, so the attribute matrix is of size 506×13 . It is important to pad this matrix with an additional column of ones, to take into account the possibility of a nonzero constant offset in the regression formula. This corresponds to the “y-intercept” in a simple one-dimensional linear regression.

Code 1.8: Multilinear regression for Boston housing data.

```
load housing.data

b = housing(:,14);           % housing values in $1000s
A = housing(:,1:13);         % other factors,
A = [A ones(size(A,1),1)];   % Pad with ones y-intercept

x = regress(b,A);
plot(b,'k-o');
```

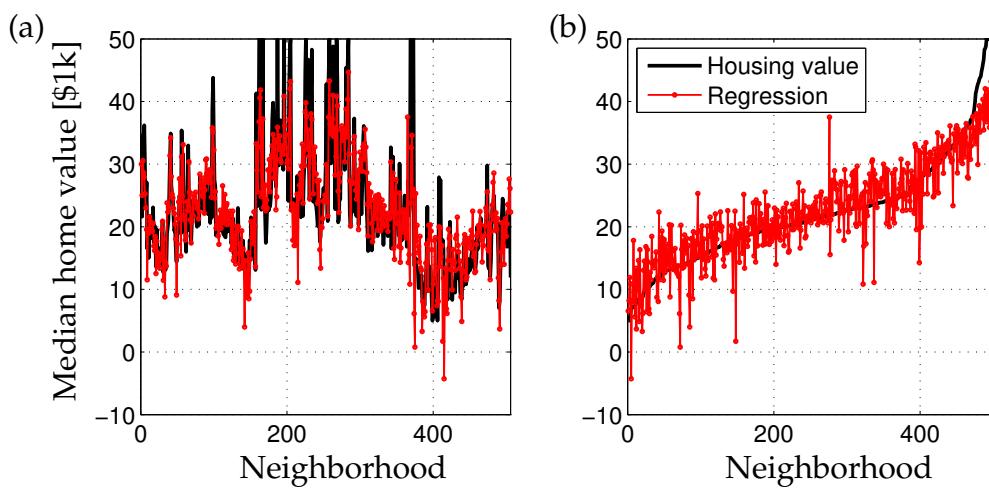


Figure 1.11: Multilinear regression of home prices using various factors. (a) Unsorted data, and (b) Data sorted by home value.

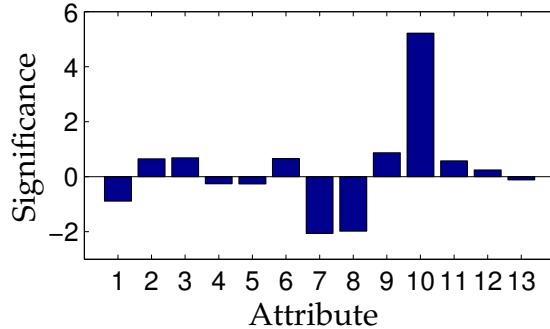


Figure 1.12: Significance of various attributes in the regression.

```

hold on, plot(A*x, 'r-o');

[b sortind] = sort(housing(:,14)); % sorted values
plot(b, 'k-o')
hold on, plot(A(sortind,:)*x, 'r-o')

```

Caution

In general, the matrix \mathbf{U} , whose columns are left-singular vectors of \mathbf{X} , is a unitary square matrix. Therefore, $\mathbf{U}^*\mathbf{U} = \mathbf{U}\mathbf{U}^* = \mathbf{I}_{n \times n}$. However, to compute the pseudo-inverse of \mathbf{X} , we must compute $\mathbf{X}^\dagger = \tilde{\mathbf{V}}\tilde{\Sigma}^{-1}\tilde{\mathbf{U}}^*$ since only $\tilde{\Sigma}$ is invertible (if all singular values are nonzero), although Σ is not invertible in general (in fact, it is generally not even square).

Until now, we have assumed that $\mathbf{X} = \tilde{\mathbf{U}}\tilde{\Sigma}\tilde{\mathbf{V}}^*$ is an exact SVD, so that the rank r includes all non-zero singular values. This guarantees that the matrix $\tilde{\Sigma}$ is invertible.

A complication arises when working with a truncated basis of left singular vectors $\tilde{\mathbf{U}}$. It is still true that $\tilde{\mathbf{U}}^*\tilde{\mathbf{U}} = \mathbf{I}_{r \times r}$, where r is the rank of \mathbf{X} . However, $\tilde{\mathbf{U}}\tilde{\mathbf{U}}^* \neq \mathbf{I}_{n \times n}$, which is easy to verify numerically on a simple example. Assuming that $\tilde{\mathbf{U}}\tilde{\mathbf{U}}^*$ is equal to the identity is one of the most common accidental misuses of the SVD⁵.

```

>> tol = 1.e-16;
>> [U,S,V] = svd(X, 'econ')
>> r = max(find(diag(S)>max(S(:))*tol));
>> invX = V(:,1:r)*S(1:r,1:r)*U(:,1:r)'; % only approximate

```

⁵The authors are not immune to this, having mistakenly used this fictional identity in an early version of [98].

1.5 Principal component analysis (PCA)

Principal components analysis (PCA) is one of the central uses of the SVD, providing a data-driven, hierarchical coordinate system to represent high-dimensional correlated data. This coordinate system involves the correlation matrices described in Sec. 1.3. Importantly, PCA pre-processes the data by mean subtraction and setting the variance to unity before performing the SVD. The geometry of the resulting coordinate system is determined by principal components (PCs) that are uncorrelated (orthogonal) to each other, but have maximal correlation with the measurements. This theory was developed in 1901 by Pearson [418], and independently by Hotelling in the 1930s [256, 257]. Jolliffe [268] provides a good reference text.

Typically, a number of measurements are collected in a single experiment, and these measurements are arranged into a row vector. The measurements may be features of an observable, such as demographic features of a specific human individual. A number of experiments are conducted, and each measurement vector is arranged as a row in a large matrix \mathbf{X} . In the example of demography, the collection of experiments may be gathered via polling. Note that this convention for \mathbf{X} , consisting of rows of features, is different than the convention throughout the remainder of this chapter, where individual feature “snapshots” are arranged as columns. However, we choose to be consistent with PCA literature in this section. The matrix will still be size $n \times m$, although it may have more rows than columns, or vice versa.

Computation

We now compute the row-wise mean $\bar{\mathbf{x}}$ (i.e., the mean of all rows), and subtract it from \mathbf{X} . The mean $\bar{\mathbf{x}}$ is given by

$$\bar{\mathbf{x}}_j = \frac{1}{n} \sum_{i=1}^n \mathbf{X}_{ij}, \quad (1.24)$$

and the mean matrix is

$$\bar{\mathbf{X}} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \bar{\mathbf{x}}. \quad (1.25)$$

Subtracting $\bar{\mathbf{X}}$ from \mathbf{X} results in the mean-subtracted data \mathbf{B} :

$$\mathbf{B} = \mathbf{X} - \bar{\mathbf{X}}. \quad (1.26)$$

The covariance matrix of the rows of \mathbf{B} is given by

$$\mathbf{C} = \frac{1}{n-1} \mathbf{B}^* \mathbf{B}. \quad (1.27)$$

The first principal component \mathbf{u}_1 is given as

$$\mathbf{u}_1 = \underset{\|\mathbf{u}_1\|=1}{\operatorname{argmax}} \mathbf{u}_1^* \mathbf{B}^* \mathbf{B} \mathbf{u}_1, \quad (1.28)$$

which is the eigenvector of $\mathbf{B}^* \mathbf{B}$ corresponding to the largest eigenvalue. Now it is clear that \mathbf{u}_1 is the left singular vector of \mathbf{B} corresponding to the largest singular value.

It is possible to obtain the principal components by computing the eigen-decomposition of \mathbf{C} :

$$\mathbf{C}\mathbf{V} = \mathbf{V}\mathbf{D}, \quad (1.29)$$

which is guaranteed to exist, since \mathbf{C} is Hermitian.

pca Command

In Matlab, there are the additional commands **pca** and **princomp** (based on **pca**) for the principal components analysis:

```
||>> [V, score, s2] = pca(X);
```

The matrix \mathbf{V} is equivalent to the \mathbf{V} matrix from the SVD of \mathbf{X} , up to sign changes of the columns. The vector $\mathbf{s2}$ contains eigenvalues of the covariance of \mathbf{X} , also known as principal component variances; these values are the squares of the singular values. The variable **score** simply contains the coordinates of each row of \mathbf{B} (the mean-subtracted data) in the principal component directions. In general, we often prefer to use the **svd** command with the various pre-processing steps described above.

Example: Noisy Gaussian data

Consider the noisy cloud of data in Fig. 1.13 (a), generated using Code 1.9. The data is generated by selecting 10,000 vectors from a two-dimensional normal distribution with zero mean and unit variance. These vectors are then scaled in the x and y directions by the values in Table 1.1 and rotated by $\pi/3$. Finally, the entire cloud of data is translated so that it has a nonzero center $\mathbf{x}_C = [2 \ 1]^T$.

Using Code 1.10, the PCA is performed and used to plot confidence intervals using multiple standard deviations, shown in Fig. 1.13 (b). The singular values, shown in Table 1.1, match the data scaling. The matrix \mathbf{U} from the SVD also closely matches the rotation matrix, up to a sign on the columns:

$$\mathbf{R}_{\pi/3} = \begin{bmatrix} 0.5 & -0.8660 \\ 0.8660 & 0.5 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} -0.4998 & -0.8662 \\ -0.8662 & 0.4998 \end{bmatrix}.$$

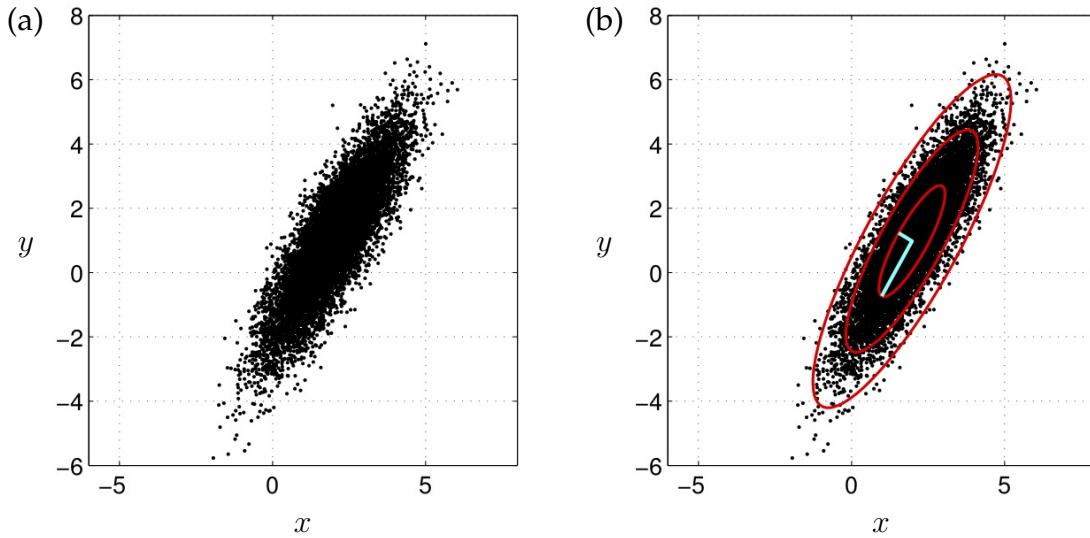


Figure 1.13: Principal components capture the variance of mean-subtracted Gaussian data (a). The first three standard deviation ellipsoids (red), and the two left singular vectors, scaled by singular values ($\sigma_1 u_1 + x_C$ and $\sigma_2 u_2 + x_C$, cyan), are shown in (b).

Code 1.9: Generation of noisy cloud of data to illustrate PCA.

```

xC = [2; 1];
sig = [2; .5];

theta = pi/3;
R = [cos(theta) -sin(theta);
      sin(theta) cos(theta)];

nPoints = 10000;
X = R*diag(sig)*randn(2,nPoints) + diag(xC)*ones(2,nPoints);
scatter(X(1,:),X(2,:),'k.', 'LineWidth', 2)

```

Table 1.1: Standard deviation of data and normalized singular values.

	σ_1	σ_2
Data	2	0.5
SVD	1.974	0.503

Code 1.10: Compute PCA and plot confidence intervals.

```
Xavg = mean(X, 2); % Compute mean
B = X - Xavg*ones(1, nPoints); % Mean-subtracted Data
[U, S, V] = svd(B/sqrt(nPoints), 'econ'); % PCA via SVD
scatter(X(1, :), X(2, :), 'k.', 'LineWidth', 2) % Plot data

theta = (0:.01:1)*2*pi;
Xstd = U*S*[cos(theta); sin(theta)]; % 1-std conf. interval
plot(Xavg(1)+Xstd(1, :), Xavg(2) + Xstd(2, :), 'r-')
plot(Xavg(1)+2*Xstd(1, :), Xavg(2) + 2*Xstd(2, :), 'r-')
plot(Xavg(1)+3*Xstd(1, :), Xavg(2) + 3*Xstd(2, :), 'r-')
```

Finally, it is also possible to compute using the **pca** command:

```
>> [V, score, s2] = pca(X);
>> norm(V*score' - B)

ans =
2.2878e-13
```

Example: Ovarian cancer data

The ovarian cancer data set, which is built into Matlab, provides a more realistic example to illustrate the benefits of PCA. This example consists of gene data for 216 patients, 121 of whom have ovarian cancer, and 95 of whom do not. For each patient, there is a vector of data containing the expression of 4000 genes. There are multiple challenges with this type of data, namely the high dimension of the data features. However, we see from Fig 1.14 that there is significant variance captured in the first few PCA modes. Said another way, the gene data is highly correlated, so that many patients have significant overlap in their gene expression. The ability to visualize patterns and correlations in high-dimensional data is an important reason to use PCA, and PCA has been widely used to find patterns in high-dimensional biological and genetic data [448].

More importantly, patients with ovarian cancer appear to cluster separately from patients without cancer when plotted in the space spanned by the first three PCA modes. This is shown in Fig. 1.15, which is generated by Code 1.11. This inherent clustering in PCA space of data by category is a foundational element of machine learning and pattern recognition. For example, we will see in Sec. 1.6 that images of different human faces will form clusters in PCA space. The use of these clusters will be explored in greater detail in Chapter 5.

Code 1.11: Compute PCA for ovarian cancer data.

```
load ovariancancer; % Load ovarian cancer data
[U, S, V] = svd(obs, 'econ');
```

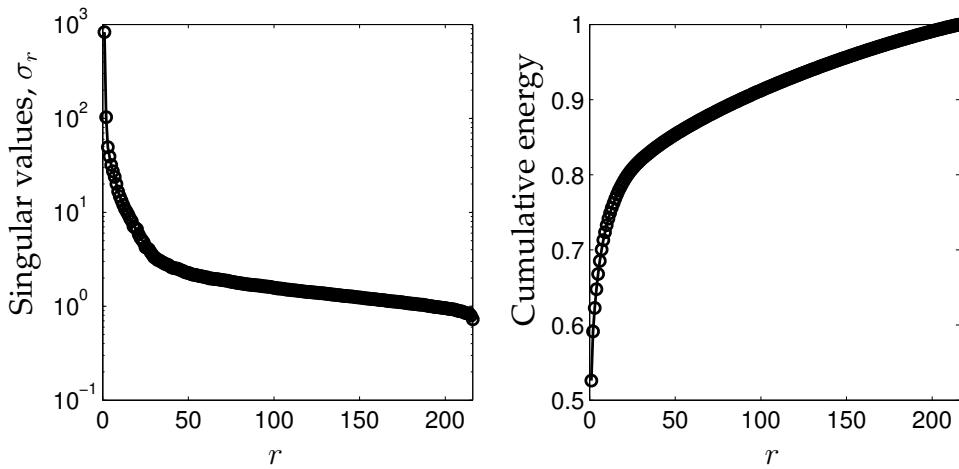


Figure 1.14: Singular values for the Ovarian cancer data.

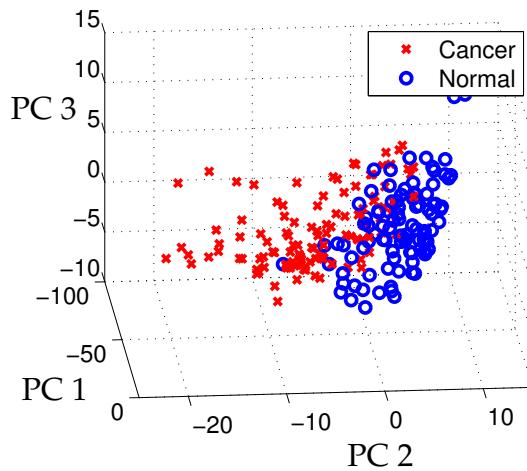


Figure 1.15: Clustering of samples that are normal and those that have cancer in the first three principal component coordinates.

```

for i=1:size(obs,1)
    x = V(:,1)'*obs(i,:)';
    y = V(:,2)'*obs(i,:)';
    z = V(:,3)'*obs(i,:)';
    if(grp{i}=='Cancer')
        plot3(x,y,z,'rx','LineWidth',2);
    else
        plot3(x,y,z,'bo','LineWidth',2);
    end
end

```



Figure 1.16: (left) A single image for each person in the Yale database, and (right) all images for a specific person. Left panel generated by Code (1.12).

1.6 Eigenfaces example

One of the most striking demonstrations of SVD/PCA is the so-called eigenfaces example. In this problem, PCA (i.e. SVD on mean-subtracted data) is applied to a large library of facial images to extract the most dominant correlations between images. The result of this decomposition is a set of *eigenfaces* that define a new coordinate system. Images may be represented in these coordinates by taking the dot product with each of the principal components. It will be shown in Chapter 5 that images of the same person tend to cluster in the eigenface space, making this a useful transformation for facial recognition and classification [510, 48]. The eigenface problem was first studied by Sirovich and Kirby in 1987 [491] and expanded on in [291]. Its application to automated facial recognition was presented by Turk and Pentland in 1991 [537].

Here, we demonstrate this algorithm using the Extended Yale Face Database B [203], consisting of cropped and aligned images [327] of 38 individuals (28 from the extended database, and 10 from the original database) under 9 poses and 64 lighting conditions⁶. Each image is 192 pixels tall and 168 pixels wide. Unlike the previous image example in Section 1.2, each of the facial images in our library have been reshaped into a large column vector with $192 \times 168 = 32,256$ elements. We use the first 36 people in the database (left panel of Fig. 1.16)

⁶The database can be downloaded at <http://vision.ucsd.edu/~iskwak/ExtYaleDatabase/ExtYaleB.html>.

as our training data for the eigenfaces example, and we hold back two people as a test set. An example of all 64 images of one specific person are shown in the right panel. These images are loaded and plotted using Code 1.12.

Code 1.12: Plot an image for each person in the Yale database (Fig. 1.16 (a))

```

load ../DATA/allFaces.mat

allPersons = zeros(n*6,m*6);           % Make an array to fit all
                                         faces
count = 1;
for i=1:6                  % 6 x 6 grid of faces
    for j=1:6
        allPersons(1+(i-1)*n:i*n,1+(j-1)*m:j*m) ...
            =reshape(faces(:,1+sum(nfaces(1:count-1))),n,m);
        count = count + 1;
    end
end
imagesc(allPersons), colormap gray

```

As mentioned before, each image is reshaped into a large column vector, and the average face is computed and subtracted from each column vector. The mean-subtracted image vectors are then stacked horizontally as columns in the data matrix \mathbf{X} , as shown in Fig. 1.17. Thus, taking the SVD of the mean-subtracted matrix \mathbf{X} results in the PCA. The columns of \mathbf{U} are the eigenfaces, and they may be reshaped back into 192×168 images. This is illustrated in Code 1.13.

Code 1.13: Compute eigenfaces on mean-subtracted data.

```

% We use the first 36 people for training data
trainingFaces = faces(:,1:sum(nfaces(1:36)));
avgFace = mean(trainingFaces,2); % size n*m by 1;

% Compute eigenfaces on mean-subtracted training data
X = trainingFaces-avgFace*ones(1,size(trainingFaces,2));
[U,S,V] = svd(X,'econ');

imagesc(reshape(avgFace,n,m)) % Plot avg face
imagesc(reshape(U(:,1),n,m)) % Plot first eigenface

```

Using the eigenface library, $\tilde{\mathbf{U}}$, obtained above, we now attempt to approximately represent an image that was not in the training data. At the beginning, we held back two individuals (the 37th and 38th people), and we now use one of their images as a test image, \mathbf{x}_{test} . We will see how well a rank- r SVD basis will approximate this image using the following projection:

$$\tilde{\mathbf{x}}_{\text{test}} = \tilde{\mathbf{U}}\tilde{\mathbf{U}}^*\mathbf{x}_{\text{test}}.$$

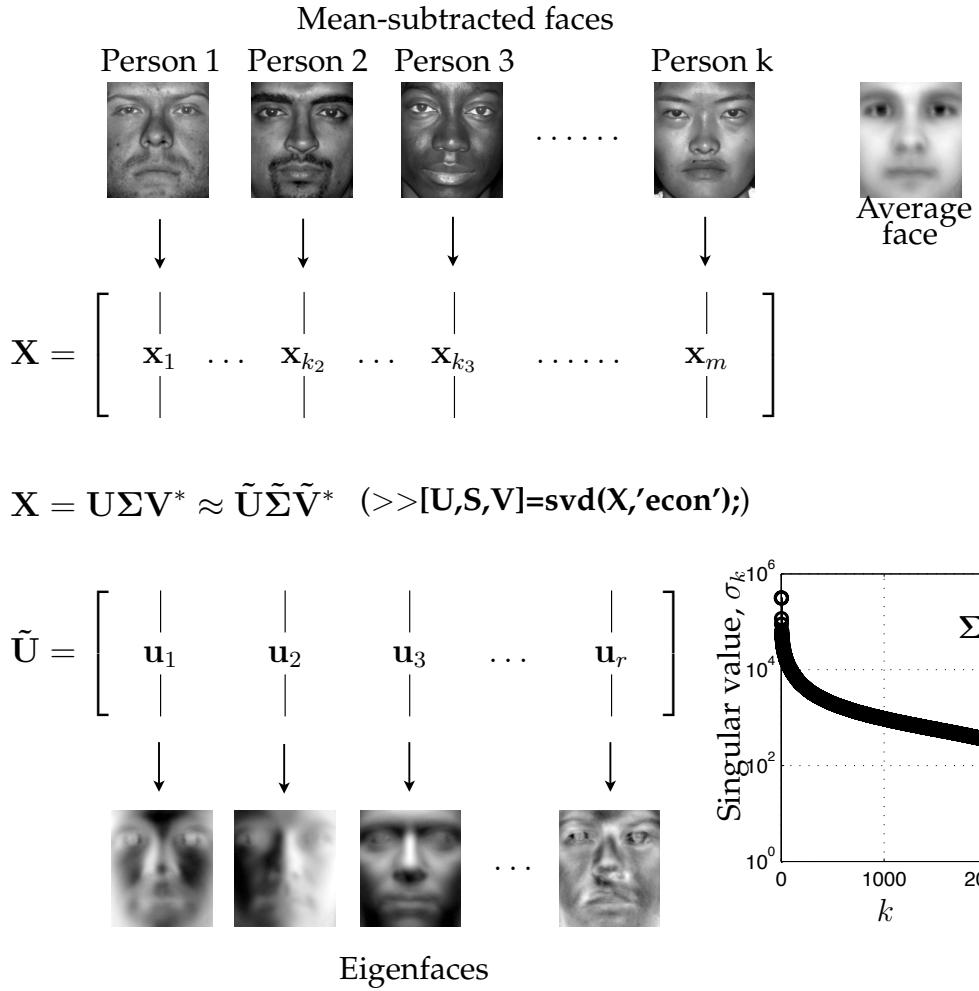


Figure 1.17: Schematic procedure to obtain eigenfaces from library of faces.

The eigenface approximation for various values of r is shown in Fig. 1.18, as computed using Code 1.14. The approximation is relatively poor for $r \leq 200$, although for $r > 400$ it converges to a passable representation of the test image.

It is interesting to note that the eigenface space is not only useful for representing human faces, but may also be used to approximate a dog (Fig. 1.19) or a cappuccino (Fig. 1.20). This is possible because the 1600 eigenfaces span a large subspace of the 32256 dimensional image space corresponding to broad, smooth, non-localized spatial features, such as cheeks, forehead, mouths, etc.

Code 1.14: Approximate test-image that was omitted from training data.

```

testFaceMS = testFace - avgFace;
for r=[25 50 100 200 400 800 1600]
    reconFace = avgFace + (U(:,1:r)*(U(:,1:r)'*testFaceMS));
    imagesc(reshape(reconFace,n,m))
end

```

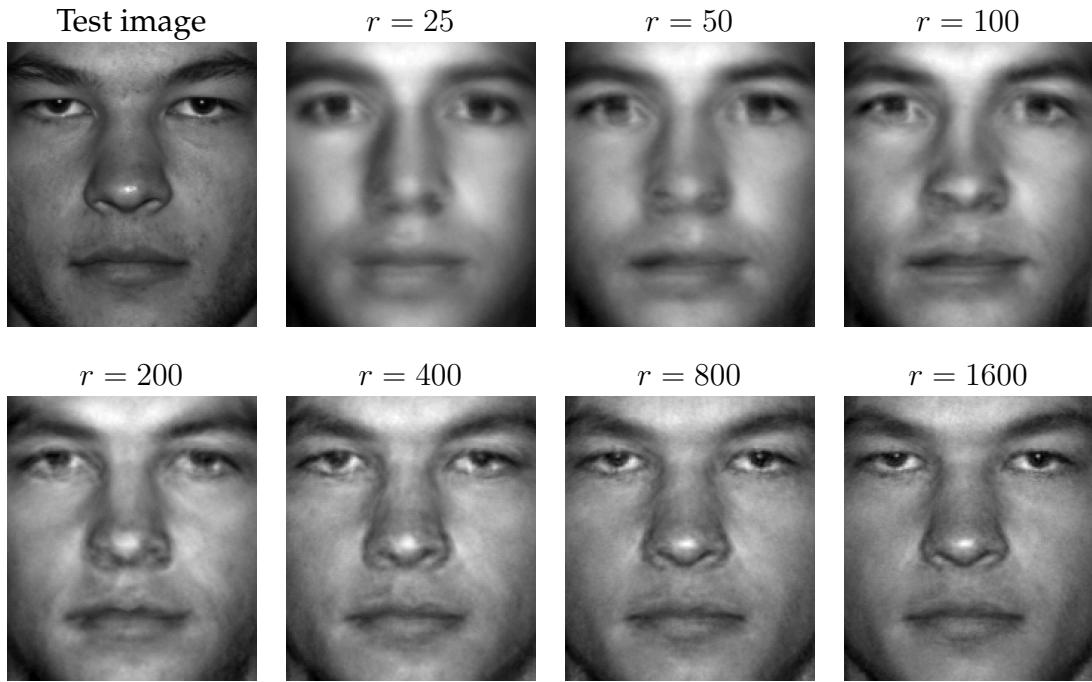


Figure 1.18: Approximate representation of test image using eigenfaces basis of various order r . Test image is not in training set.

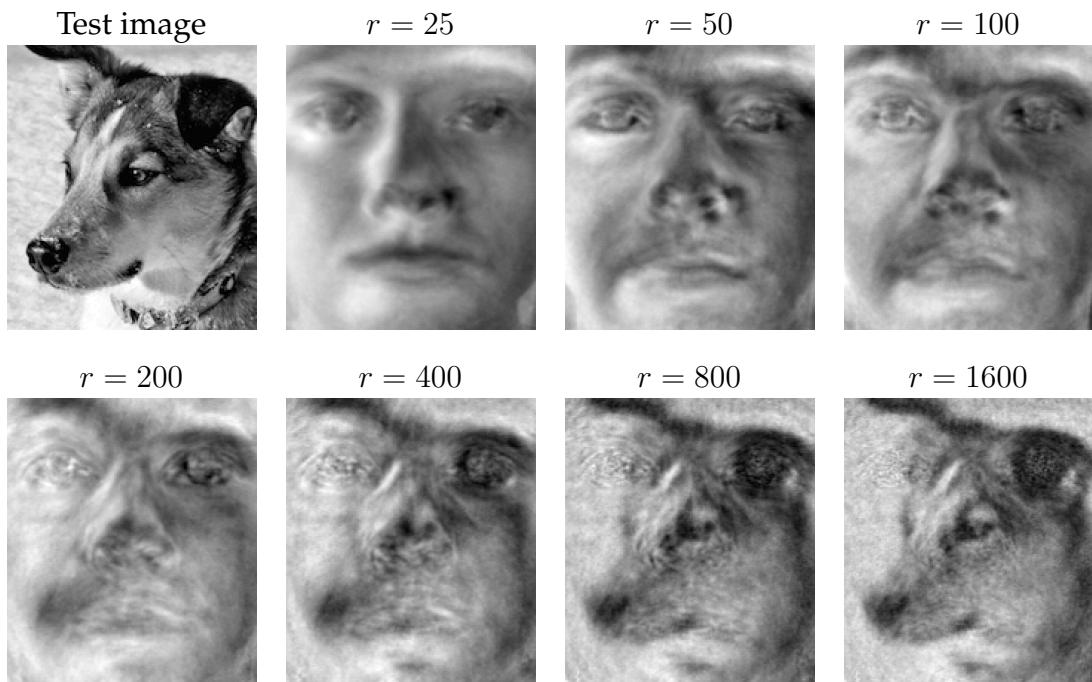


Figure 1.19: Approximate representation of an image of a dog using eigenfaces.

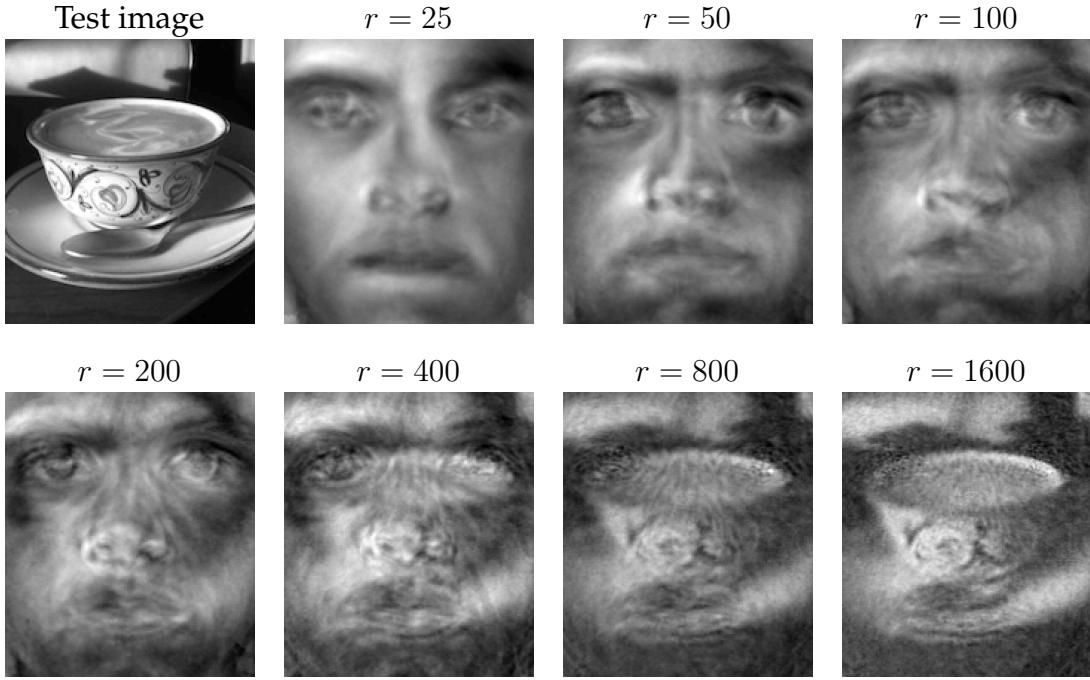


Figure 1.20: Approximate representation of a cappuccino using eigenfaces.

We further investigate the use of the eigenfaces as a coordinate system, defining an eigenface space. By projecting an image \mathbf{x} onto the first r PCA modes, we obtain a set of coordinates in this space: $\tilde{\mathbf{x}} = \tilde{\mathbf{U}}^* \mathbf{x}$. Some principal components may capture the most common features shared among all human faces, while other principal components will be more useful for distinguishing between individuals. Additional principal components may capture differences in lighting angles. Figure 1.21 shows the coordinates of all 64 images of two individuals projected onto the 5th and 6th principal components, generated by Code 1.15. Images of the two individuals appear to be well-separated in these coordinates. This is the basis for image recognition and classification in Chapter 5.

Code 1.15: Project images for two specific people onto the 5th and 6th eigenfaces to illustrate the potential for automated classification.

```

P1num = 2; % Person number 2
P2num = 7; % Person number 7

P1 = faces(:, 1+sum(nfaces(1:P1num-1)) : sum(nfaces(1:P1num)));
P2 = faces(:, 1+sum(nfaces(1:P2num-1)) : sum(nfaces(1:P2num)));

P1 = P1 - avgFace*ones(1, size(P1, 2));
P2 = P2 - avgFace*ones(1, size(P2, 2));

```

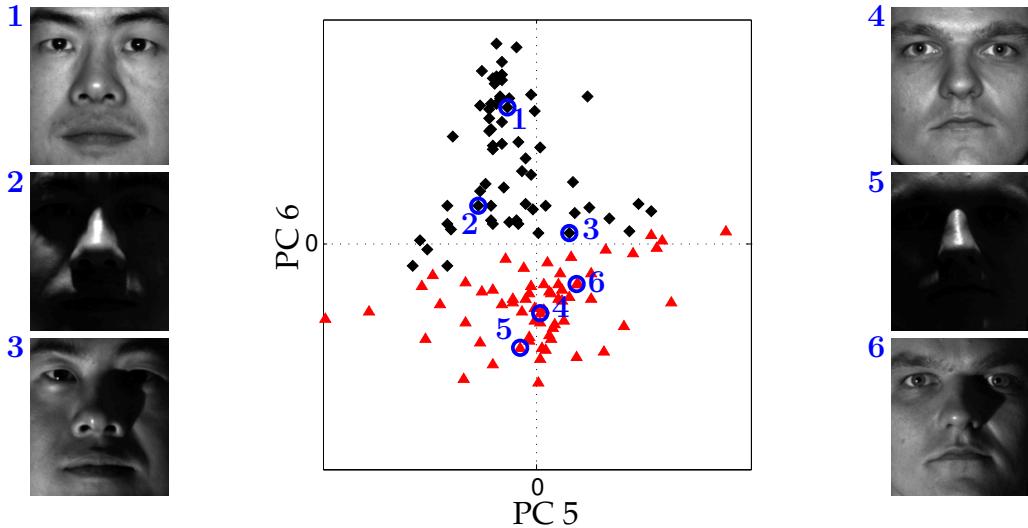


Figure 1.21: Projection of all images from two individuals onto the 5th and 6th PCA modes. Projected images of the first individual are indicated with black diamonds, and projected images of the second individual are indicated with red triangles. Three examples from each individual are circled in blue, and the corresponding image is shown.

```

PCAmodes = [5 6];      % Project onto PCA modes 5 and 6
PCACoordsP1 = U(:, PCAmodes)' * P1;
PCACoordsP2 = U(:, PCAmodes)' * P2;

plot(PCACoordsP1(1, :), PCACoordsP1(2, :), 'kd')
plot(PCACoordsP2(1, :), PCACoordsP2(2, :), 'r^')

```

1.7 Truncation and alignment

Deciding how many singular values to keep, i.e. where to truncate, is one of the most important and contentious decisions when using the SVD. There are many factors, including specifications on the desired rank of the system, the magnitude of noise, and the distribution of the singular values. Often, one truncates the SVD at a rank r that captures a pre-determined amount of the variance or energy in the original data, such as 90% or 99% truncation. Although crude, this technique is commonly used. Other techniques involve identifying “elbows” or “knees” in the singular value distribution, which may denote the transition from singular values that represent important patterns from those that represent noise. Truncation may be viewed as a hard threshold on singu-

lar values, where values larger than a threshold τ are kept, while remaining singular values are truncated. Recent work by Gavish and Donoho [200] provides an optimal truncation value, or hard threshold, under certain conditions, providing a principled approach to obtaining low-rank matrix approximations using the SVD.

In addition, the alignment of data significantly impacts the rank of the SVD approximation. The SVD essentially relies on a separation of variables between the columns and rows of a data matrix. In many situations, such as when analyzing traveling waves or misaligned data, this assumption breaks down, resulting in an artificial rank inflation.

Optimal hard threshold

A recent theoretical breakthrough determines the *optimal* hard threshold τ for singular value truncation under the assumption that a matrix has a low-rank structure contaminated with Gaussian white noise [200]. This work builds on a significant literature surrounding various techniques for hard and soft thresholding of singular values. In this section, we summarize the main results and demonstrate the thresholding on various examples. For more details, see [200].

First, we assume that the data matrix \mathbf{X} is the sum of an underlying low-rank, or approximately low-rank, matrix \mathbf{X}_{true} and a noise matrix $\mathbf{X}_{\text{noise}}$:

$$\mathbf{X} = \mathbf{X}_{\text{true}} + \gamma \mathbf{X}_{\text{noise}}. \quad (1.30)$$

The entries of $\mathbf{X}_{\text{noise}}$ are assumed to be independent, identically distributed (i.i.d.) Gaussian random variables with zero mean and unit variance. The magnitude of the noise is characterized by γ , which deviates from the notation in [200]⁷.

When the noise magnitude γ is known, there are closed-form solutions for the optimal hard threshold τ :

1. If $\mathbf{X} \in \mathbb{R}^{n \times n}$ is square, then

$$\tau = (4/\sqrt{3})\sqrt{n}\gamma. \quad (1.31)$$

2. If $\mathbf{X} \in \mathbb{R}^{n \times m}$ is rectangular and $m \ll n$, then the constant $4/\sqrt{3}$ is replaced by a function of the aspect ratio $\beta = m/n$:

$$\tau = \lambda(\beta)\sqrt{n}\gamma, \quad (1.32)$$

$$\lambda(\beta) = \left(2(\beta + 1) + \frac{8\beta}{(\beta + 1) + (\beta^2 + 14\beta + 1)^{1/2}} \right)^{1/2}. \quad (1.33)$$

⁷In [200], σ is used to denote standard deviation and y_k denotes the k^{th} singular value.

Note that this expression reduces to (1.31) when $\beta = 1$. If $n \ll m$, then $\beta = n/m$.

When the noise magnitude γ is unknown, which is more typical in real-world applications, then it is possible to estimate the noise magnitude and scale the distribution of singular values by using σ_{med} , the *median* singular value. In this case, there is no closed-form solution for τ , and it must be approximated numerically.

3. For unknown noise γ , and a rectangular matrix $\mathbf{X} \in \mathbb{R}^{n \times m}$, the optimal hard threshold is given by

$$\tau = \omega(\beta)\sigma_{\text{med}}. \quad (1.34)$$

Here, $\omega(\beta) = \lambda(\beta)/\mu_\beta$, where μ_β is the solution to the following problem:

$$\int_{(1-\beta)^2}^{\mu_\beta} \frac{[((1+\sqrt{\beta})^2 - t)(t - (1-\sqrt{\beta})^2)]^{1/2}}{2\pi t} dt = \frac{1}{2}.$$

Solutions to the expression above must be approximated numerically. Fortunately [200] has a Matlab code supplement⁸ [152] to approximate μ_β .

The new method of optimal hard thresholding works remarkably well, as demonstrated on the examples below.

Example 1: Toy problem

In the first example, shown in Fig. 1.22, we artificially construct a rank-2 matrix (Code 1.16) and we contaminate the signal with Gaussian white noise (Code 1.17). A de-noised and dimensionally reduced matrix is then obtained using the threshold from (1.31) (Code 1.18), as well as using a 90% energy truncation (Code 1.19). It is clear that the hard threshold is able to filter the noise more effectively. Plotting the singular values (Code 1.20) in Fig. 1.23, it is clear that there are two values that are above threshold.

Code 1.16: Compute the underlying low-rank signal. (Fig. 1.22 (a))

```
clear all, close all, clc
t = (-3:.01:3)';
Utrue = [cos(17*t).*exp(-t.^2) sin(11*t)];
Strue = [2 0; 0 .5];
```

⁸<http://purl.stanford.edu/vg705qn9070>

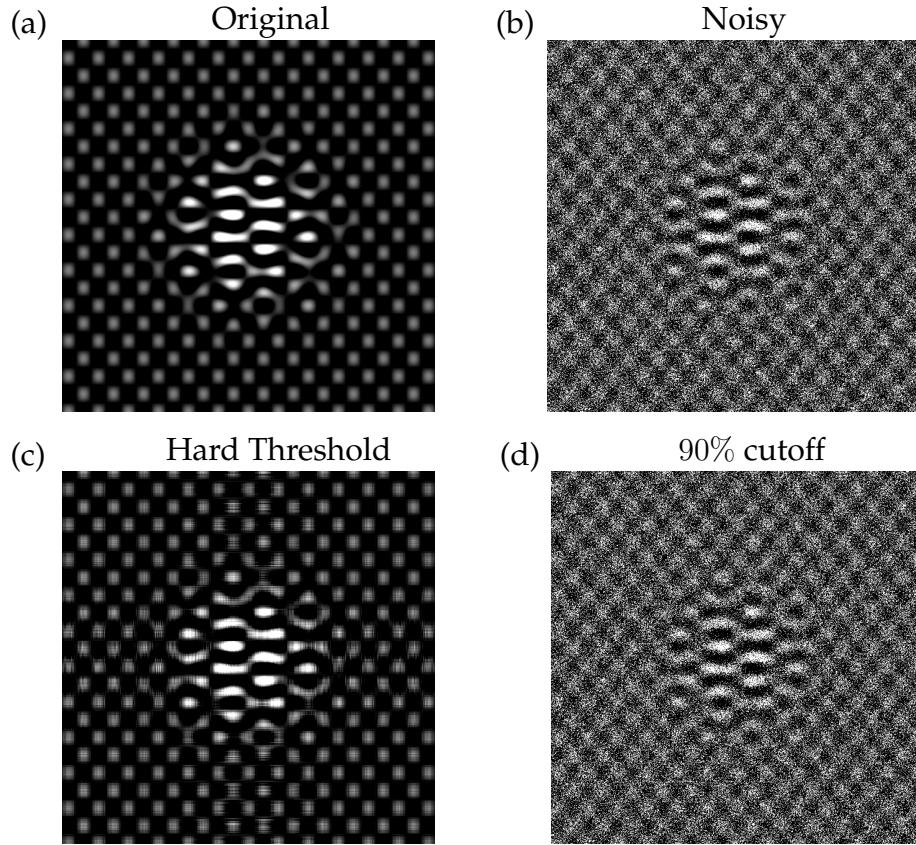


Figure 1.22: Underlying rank 2 matrix (a), matrix with noise (b), clean matrix after optimal hard threshold $(4/\sqrt{3})\sqrt{n}\sigma$ (c), and truncation based on 90% energy (d).

```

||| Vtrue = [sin(5*t).*exp(-t.^2) cos(13*t)] ;
|||
||| X = Utrue*Strue*Vtrue';
||| figure, imshow(X);

```

Code 1.17: Contaminate the signal with noise. (Fig. 1.22 (b))

```

||| sigma = 1;
||| Xnoisy = X+sigma*randn(size(X));
||| figure, imshow(Xnoisy);

```

Code 1.18: Truncate using optimal hard threshold. (Fig. 1.22 (c))

```

||| [U,S,V] = svd(Xnoisy);
|||
||| N = size(Xnoisy,1);
||| cutoff = (4/sqrt(3))*sqrt(N)*sigma; % Hard threshold
||| r = max(find(diag(S)>cutoff)); % Keep modes w/ sig > cutoff

```

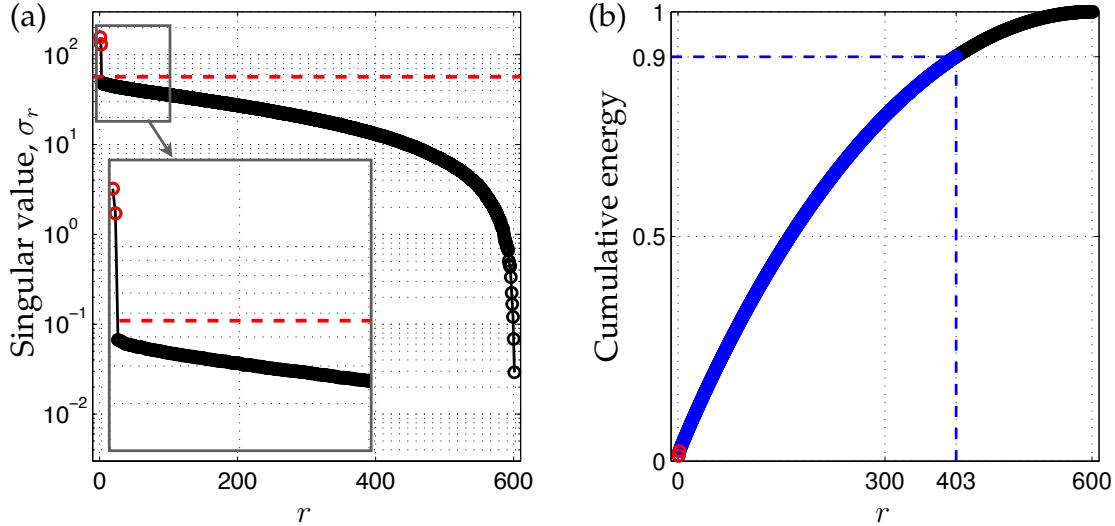


Figure 1.23: Singular values σ_r (a) and cumulative energy in first r modes (b). The optimal hard threshold $\tau = (4/\sqrt{3})\sqrt{n}\sigma$ is shown as a red dashed line (---), and the 90% cutoff is shown as a blue dashed line (- -). For this case, $n = 600$ and $\sigma = 1$ so that the optimal cutoff is approximately $\tau = 56.6$.

```
Xclean = U(:,1:r)*S(1:r,1:r)*V(:,1:r)';
figure, imshow(Xclean)
```

Code 1.19: Truncate using 90% energy criterion. (Fig. 1.22 (d))

```
cds = cumsum(diag(S))./sum(diag(S)); % Cumulative energy
r90 = min(find(cds>0.90)); % Find r to capture 90% energy

X90 = U(:,1:r90)*S(1:r90,1:r90)*V(:,1:r90)';
figure, imshow(X90)
```

Code 1.20: Plot singular values for hard threshold example. (Fig. 1.23)

```
semilogy(diag(S),'-ok','LineWidth',1.5), hold on, grid on
semilogy(diag(S(1:r,1:r)),'or','LineWidth',1.5)
```

Example 2: Eigenfaces

In the second example, we revisit the eigenfaces problem from Section 1.6. This provides a more typical example, since the data matrix X is rectangular, with aspect ratio $\beta = 3/4$, and the noise magnitude is unknown. It is also not clear that the data is contaminated with white noise. Nonetheless, the method determines a threshold τ , above which columns of U appear to have strong facial features, and below which columns of U consist mostly of noise, shown in Fig. 1.24.

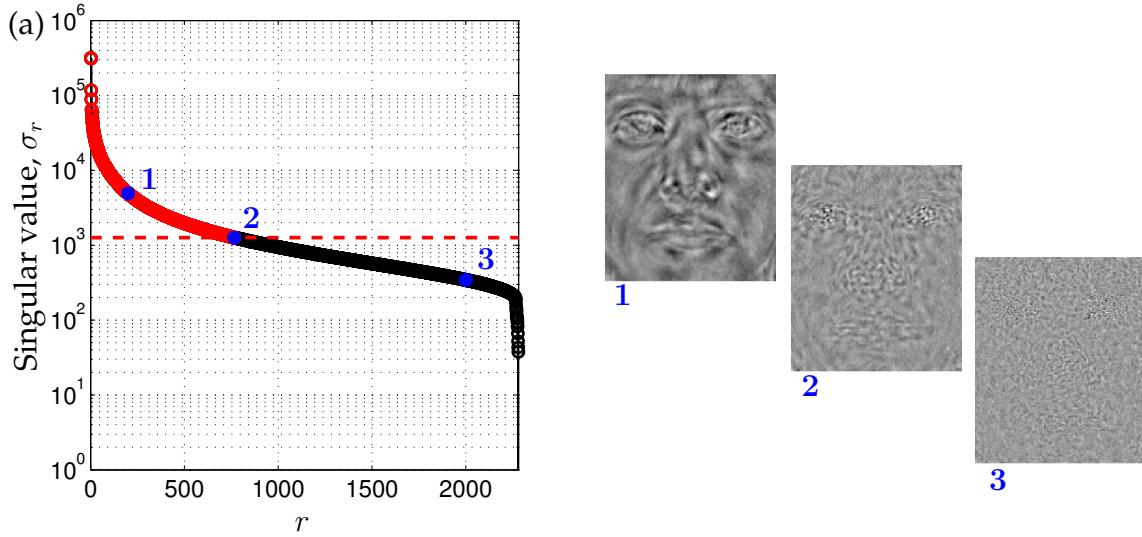


Figure 1.24: Hard thresholding for eigenfaces example.

Importance of data alignment

Here, we discuss common pitfalls of the SVD associated with misaligned data. The following example is designed to illustrate one of the central weaknesses of the SVD for dimensionality reduction and coherent feature extraction in data. Consider a matrix of zeros with a rectangular sub-block consisting of ones. As an image, this would look like a white rectangle placed on a black background (see Fig. 1.25 (a)). If the rectangle is perfectly aligned with the x - and y - axes of the figure, then the SVD is simple, having only one nonzero singular value σ_1 (see Fig. 1.25 (c)) and corresponding singular vectors u_1 and v_1 that define the width and height of the white rectangle.

When we begin to rotate the inner rectangle so that it is no longer aligned with the image axes, additional non-zero singular values begin to appear in the spectrum (see Figs. 1.25 (b,d) and 1.26).

Code 1.21: Compute the SVD for a well-aligned and rotated square (Fig. 1.25).

```

n = 1000;      % 1000 x 1000 square
X = zeros(n,n);
X(n/4:3*n/4, n/4:3:n/4) = 1;
imshow(X);

Y = imrotate(X,10,'bicubic');    % rotate 10 degrees
Y = Y - Y(1,1);
nY = size(Y,1);
startind = floor((nY-n)/2);
Xrot = Y(startind:startind+n-1, startind:startind+n-1);
imshow(Xrot);

```

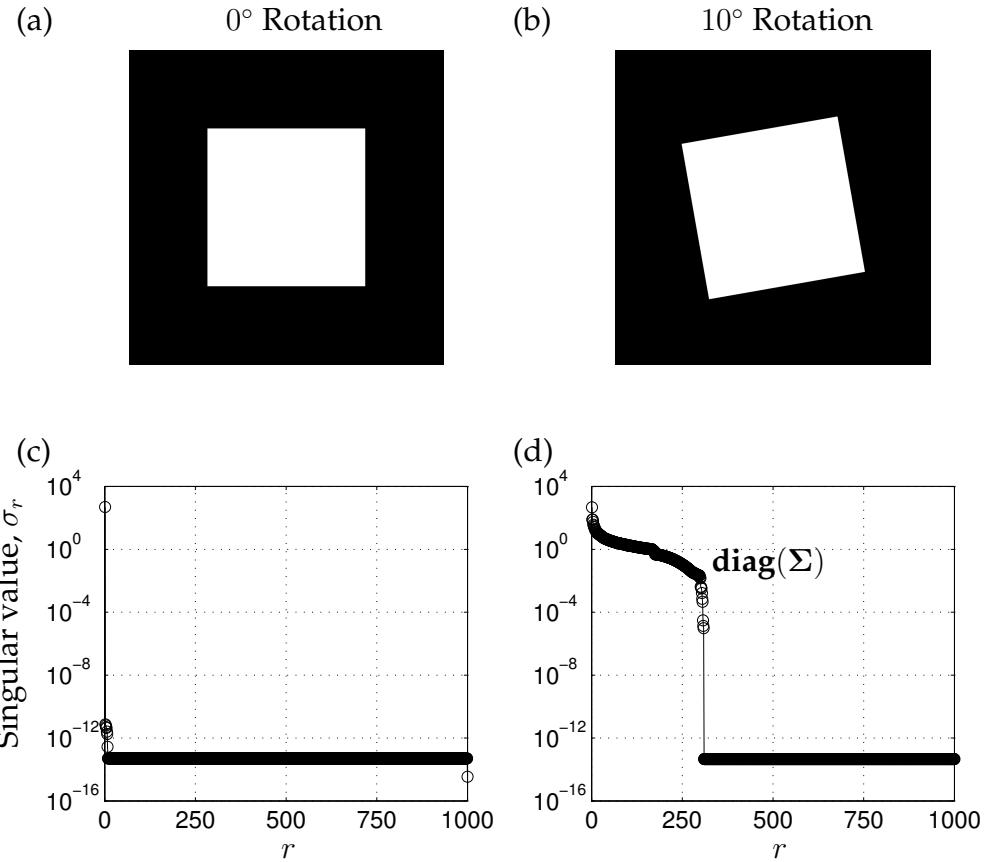


Figure 1.25: A data matrix consisting of ones with a square sub-block of zeros (a), and its SVD spectrum (c). If we rotate the image by 10°, as in (b), the SVD spectrum becomes significantly more complex (d).

```

||| [U,S,V] = svd(X);      % SVD well-aligned square
||| [U,S,V] = svd(Xrot);   % SVD rotated square
||| semilogy(diag(S), '-ko')
||| semilogy(diag(S), '-ko')

```

The reason that this example breaks down is that the SVD is fundamentally *geometric*, meaning that it depends on the coordinate system in which the data is represented. As we have seen earlier, the SVD is only generically invariant to unitary transformations, meaning that the transformation preserves the inner product. This fact may be viewed as both a strength and a weakness of the method. First, the dependence of SVD on the inner product is essential for the various useful geometric interpretations. Moreover, the SVD has meaningful units and dimensions. However, this makes the SVD sensitive to the alignment of the data. In fact, the SVD rank explodes when objects in the columns translate, rotate, or scale, which severely limits its use for data that has not been

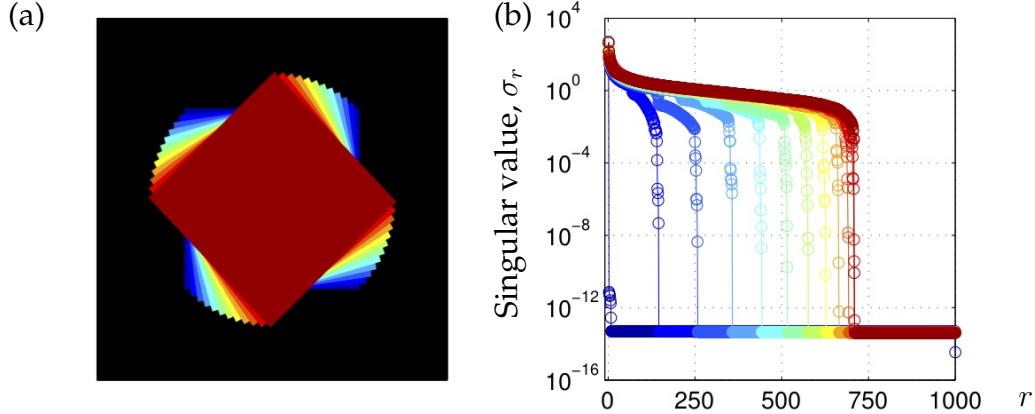


Figure 1.26: A data matrix consisting of zeros with a square sub-block of ones at various rotations (a), and the corresponding SVD spectrum, $\text{diag}(S)$, (b).

heavily pre-processed.

For instance, the eigenfaces example was built on a library of images that had been meticulously cropped, centered, and aligned according to a stencil. Without taking these important pre-processing steps, the features and clustering performance would be underwhelming.

The inability of the SVD to capture translations and rotations of the data is a major limitation. For example, the SVD is still the method of choice for the low-rank decomposition of data from partial differential equations (PDEs), as will be explored in Chapters 11 and 12. However, the SVD is fundamentally a data-driven separation of variables, which we know will not work for many types of PDE, for example those that exhibit traveling waves. Generalized decompositions that retain the favorable properties and are applicable to data with symmetries is a significant open challenge in the field.

Code 1.22: SVD for a square rotated through various angles (Fig. 1.26).

```

nAngles = 12; % sweep through 12 angles, from 0:4:44
Xrot = X;
for j=2:nAngles
    Y = imrotate(X, (j-1)*4, 'bicubic'); % rotate (j-1)*4
    startind = floor((size(Y,1)-n)/2);
    Xrot1 = Y(startind:startind+n-1, startind:startind+n-1);
    Xrot2 = Xrot1 - Xrot1(1,1);
    Xrot2 = Xrot2/max(Xrot2(:));
    Xrot(Xrot2>.5) = j;

    [U,S,V] = svd(Xrot1);
    subplot(1,2,1), imagesc(Xrot), colormap([0 0 0; cm])
    subplot(1,2,2), semilogy(diag(S),'-o','color',cm(j,:))
end

```

1.8 Randomized singular value decomposition

The accurate and efficient decomposition of large data matrices is one of the cornerstones of modern computational mathematics and data science. In many cases, matrix decompositions are explicitly focused on extracting dominant low-rank structure in the matrix, as illustrated throughout the examples in this chapter. Recently, it has been shown that if a matrix \mathbf{X} has low-rank structure, then there are extremely efficient matrix decomposition algorithms based on the theory of random sampling; this is closely related to the idea of sparsity and the high-dimensional geometry of sparse vectors, which will be explored in Chapter 3. These so-called *randomized* numerical methods have the potential to transform computational linear algebra, providing accurate matrix decompositions at a fraction of the cost of deterministic methods. Moreover, with increasingly vast measurements (e.g., from 4K and 8K video, internet of things, etc.), it is often the case that the *intrinsic* rank of the data does not increase appreciable, even though the dimension of the ambient measurement space grows. Thus, the computational savings of randomized methods will only become more important in the coming years and decades with the growing deluge of data.

Randomized linear algebra

Randomized linear algebra is a much more general concept than the treatment presented here for the SVD. In addition to the randomized SVD [464, 371], randomized algorithms have been developed for principal component analysis [454, 228], the pivoted LU decomposition [485], the pivoted QR decomposition [162], and the dynamic mode decomposition [175]. Most randomized matrix decompositions can be broken into a few common steps, as described here. There are also several excellent surveys on the topic [354, 229, 334, 177]. We assume that we are working with tall-skinny matrices, so that $n > m$, although the theory readily generalizes to short-fat matrices.

Step 0: Identify a target rank, $r < m$.

Step 1: Using random projections \mathbf{P} to sample the column space, find a matrix \mathbf{Q} whose columns approximate the column space of \mathbf{X} , i.e., so that $\mathbf{X} \approx \mathbf{Q}\mathbf{Q}^*\mathbf{X}$.

Step 2: Project \mathbf{X} onto the \mathbf{Q} subspace, $\mathbf{Y} = \mathbf{Q}^*\mathbf{X}$, and compute the matrix decomposition on \mathbf{Y} .

Step 3: Reconstruct high dimensional modes $\mathbf{U} = \mathbf{Q}\mathbf{U}_{\mathbf{Y}}$ using \mathbf{Q} and the modes computed from \mathbf{Y} .

Randomized SVD algorithm

Over the past two decades, there have been several randomized algorithms proposed to compute a low-rank SVD, including the *Monte Carlo* SVD [190] and more robust approaches based on random projections [464, 335, 371]. These methods were improved by incorporating structured sampling matrices for faster matrix multiplications [559]. Here, we use the randomized SVD algorithm of Halko, Martinsson, and Tropp [229], which combined and expanded on these previous algorithms, providing favorable error bounds. Additional analysis and numerical implementation details are found in Voronin and Martinsson [544]. A schematic of the rSVD algorithm is shown in Fig. 1.27.

Step 1: We construct a random projection matrix $\mathbf{P} \in \mathbb{R}^{m \times r}$ to sample the column space of $\mathbf{X} \in \mathbb{R}^{n \times m}$:

$$\mathbf{Z} = \mathbf{XP}. \quad (1.35)$$

The matrix \mathbf{Z} may be much smaller than \mathbf{X} , especially for low-rank matrices with $r \ll m$. It is highly unlikely that a random projection matrix \mathbf{P} will project out important components of \mathbf{X} , and so \mathbf{Z} approximates the column space of \mathbf{X} with high probability. Thus, it is possible to compute the low-rank QR decomposition of \mathbf{Z} to obtain an orthonormal basis for \mathbf{X} :

$$\mathbf{Z} = \mathbf{QR}. \quad (1.36)$$

Step 2: With the low-rank basis \mathbf{Q} , we may project \mathbf{X} into a smaller space:

$$\mathbf{Y} = \mathbf{Q}^* \mathbf{X}. \quad (1.37)$$

It also follows that $\mathbf{X} \approx \mathbf{Q}\mathbf{Y}$, with better agreement when the singular values σ_k decay rapidly for $k > r$.

It is now possible to compute the singular value decomposition on \mathbf{Y} :

$$\mathbf{Y} = \mathbf{U}_\mathbf{Y} \Sigma \mathbf{V}^*. \quad (1.38)$$

Because \mathbf{Q} is a orthonormal and approximates the column space of \mathbf{X} , the matrices Σ and \mathbf{V} are the same for \mathbf{Y} and \mathbf{X} , as discussed in Sec. 1.3.

Step 3: Finally, it is possible to reconstruct the high-dimensional left singular vectors \mathbf{U} using $\mathbf{U}_\mathbf{Y}$ and \mathbf{Q} :

$$\mathbf{U} = \mathbf{Q} \mathbf{U}_\mathbf{Y}. \quad (1.39)$$

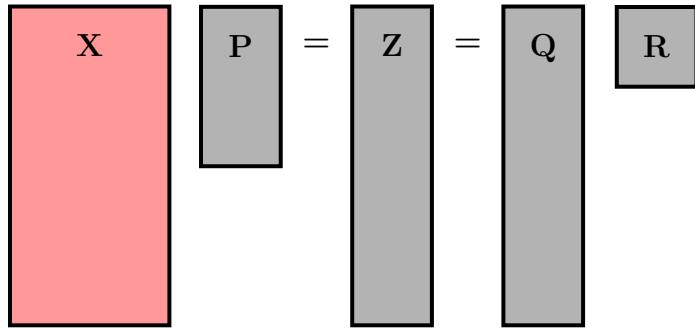
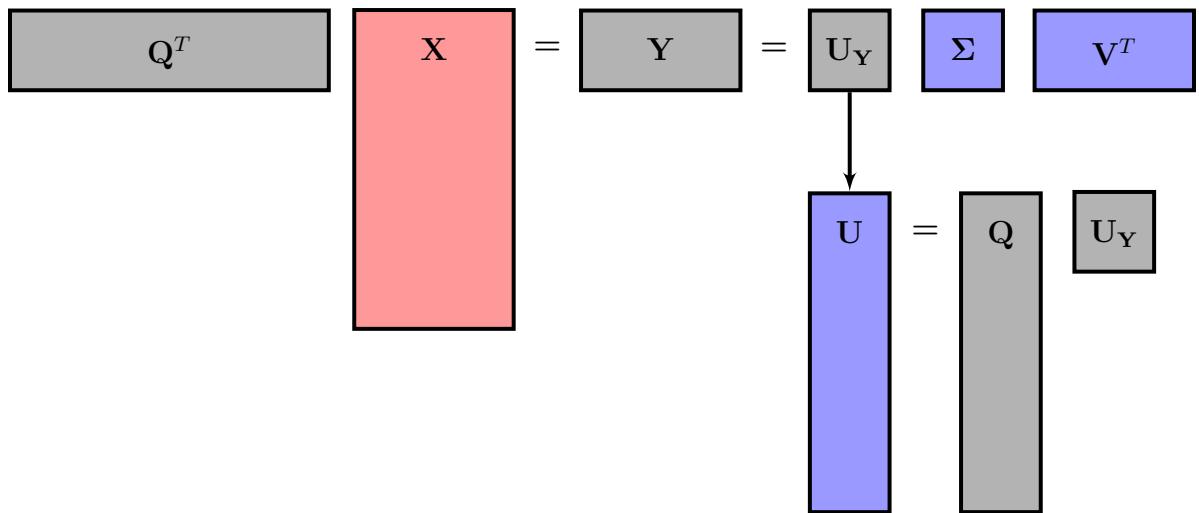
Step 1**Step 2**

Figure 1.27: Schematic of randomized SVD algorithm. The high-dimensional data \mathbf{X} is depicted in red, intermediate steps in gray, and the outputs in blue. This algorithm requires two passes over \mathbf{X} .

Oversampling

Most matrices \mathbf{X} do not have an exact low-rank structure, given by r modes. Instead, there are nonzero singular values σ_k for $k > r$, and the sketch \mathbf{Z} will not exactly span the column space of \mathbf{X} . In general, increasing the number of columns in \mathbf{P} from r to $r + p$, significantly improves results, even with p adding around 5 or 10 columns [370]. This is known as *oversampling*, and increasing p decreases the variance of the singular value spectrum of the sketched matrix.

Power iterations

A second challenge in using randomized algorithms is when the singular value spectrum decays slowly, so that the remaining truncated singular values contain significant variance in the data \mathbf{X} . In this case, it is possible to preprocess \mathbf{X} through q power iterations [454, 229, 224] to create a new matrix $\mathbf{X}^{(q)}$ with a more rapid singular value decay:

$$\mathbf{X}^{(q)} = (\mathbf{X}\mathbf{X}^*)^q \mathbf{X}. \quad (1.40)$$

Power iterations dramatically improve the quality of the randomized decomposition, as the singular value spectrum of $\mathbf{X}^{(q)}$ decays more rapidly:

$$\mathbf{X}^{(q)} = \mathbf{U}\Sigma^{2q-1}\mathbf{V}^*. \quad (1.41)$$

However, power iterations are expensive, requiring q additional passes through the data \mathbf{X} . In some extreme examples, the data in \mathbf{X} may be stored in a distributed architecture, so that every additional pass adds considerable expense.

Guaranteed error bounds

One of the most important properties of the randomized SVD is the existence of tunable error bounds, that are explicit functions of the singular value spectrum, the desired rank r , the oversampling parameter p and the number of power iterations q . The best attainable error bound for a deterministic algorithm is:

$$\|\mathbf{X} - \mathbf{Q}\mathbf{Y}\|_2 \geq \sigma_{r+1}(\mathbf{X}). \quad (1.42)$$

In other words, the approximation with the best possible rank- r subspace \mathbf{Q} will have error greater than or equal to the next truncated singular value of \mathbf{X} . For randomized methods, it is possible to bound the *expectation* of the error:

$$\mathbb{E}(\|\mathbf{X} - \mathbf{Q}\mathbf{Y}\|_2) \leq \left(1 + \sqrt{\frac{r}{p-1}} + \frac{e\sqrt{r+p}}{p}\sqrt{m-r}\right)^{\frac{1}{2q+1}} \sigma_{k+1}(\mathbf{X}), \quad (1.43)$$

where e is Euler's number.

Choice of random matrix \mathbf{P}

There are several suitable choices of the random matrix \mathbf{P} . Gaussian random projections (e.g., the elements of \mathbf{P} are i.i.d. Gaussian random variables) are frequently used because of favorable mathematical properties and the richness of information extracted in the sketch \mathbf{Z} . In particular, it is very unlikely that a Gaussian random matrix \mathbf{P} will be chosen *badly* so as to project out important information in \mathbf{X} . However, Gaussian projections are expensive to generate,

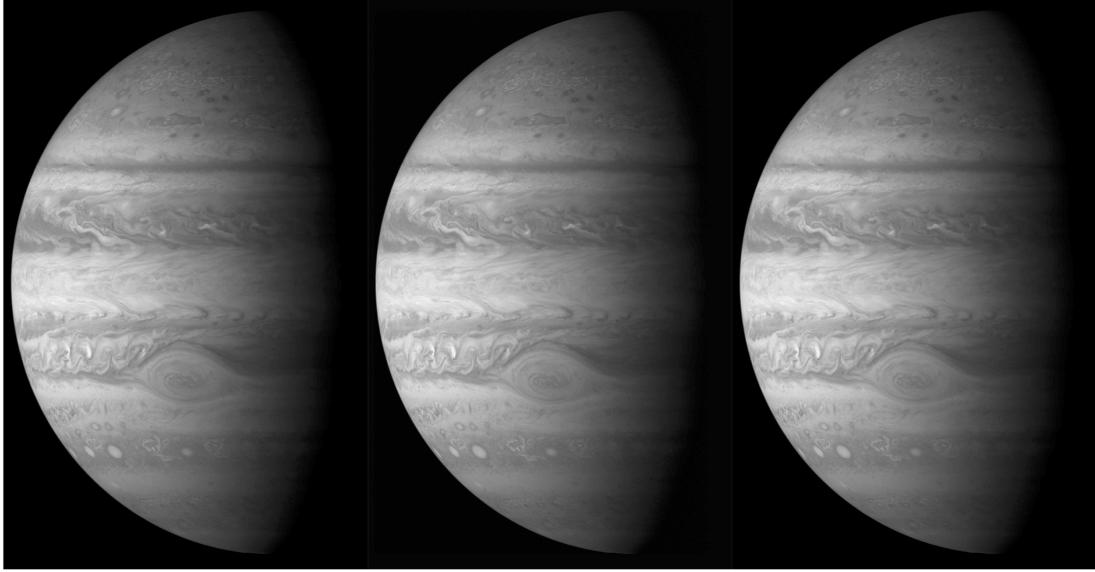


Figure 1.28: Original high-resolution (left) and rank-400 approximations from the SVD (middle) and rSVD (right).

store, and compute. Uniform random matrices are also frequently used, and have similar limitations. There are several alternatives, such as Rademacher matrices, where the entries can be $+1$ or -1 with equal probability [532]. Structured random projection matrices may provide efficient sketches, reducing computational costs to $\mathcal{O}(nm \log(r))$ [559]. Yet another choice is a sparse projection matrix \mathbf{P} , which improves storage and computation, but at the cost of including less information in the sketch. In the extreme case, when even a single pass over the matrix \mathbf{X} is prohibitively expensive, the matrix \mathbf{P} may be chosen as random columns of the $m \times m$ identity matrix, so that it randomly selects columns of \mathbf{X} for the sketch \mathbf{Z} . This is the fastest option, but should be used with caution, as information may be lost if the structure of \mathbf{X} is highly localized in a subset of columns, which may be lost by column sampling.

Example of randomized SVD

To demonstrate the randomized SVD algorithm, we will decompose a high resolution image. This particular implementation is only for illustrative purposes, as it has not been optimized for speed, data transfer, or accuracy. In practical applications, care should be taken [229, 177].

Code 1.23 computes the randomized SVD of a matrix \mathbf{X} , and Code 1.24 uses this function to obtain a rank-400 approximation to a high-resolution image, shown in Fig. 1.28.

Code 1.23: Randomized SVD algorithm.

```

function [U,S,V] = rsvd(X,r,q,p);

% Step 1: Sample column space of X with P matrix
ny = size(X,2);
P = randn(ny,r+p);
Z = X*P;
for k=1:q
    Z = X*(X'*Z);
end
[Q,R] = qr(Z,0);

% Step 2: Compute SVD on projected Y=Q'*X;
Y = Q'*X;
[Uy,S,V] = svd(Y,'econ');
U = Q*Uy;

```

Code 1.24: Compute the randomized SVD of high-resolution image.

```

clear all, close all, clc
A=imread('jupiter.jpg');
X=double(rgb2gray(A));
[U,S,V] = svd(X,'econ'); % Deterministic SVD

r = 400; % Target rank
q = 1; % Power iterations
p = 5; % Oversampling parameter
[rU,rS,rV] = rsvd(X,r,q,p); % Randomized SVD

%% Reconstruction
XSVD = U(:,1:r)*S(1:r,1:r)*V(:,1:r)'; % SVD approx.
errSVD = norm(X-XSVD,2)/norm(X,2);
XrSVD = rU(:,1:r)*rS(1:r,1:r)*rV(:,1:r)'; % rSVD approx.
errrSVD = norm(X-XrSVD,2)/norm(X,2);

```

1.9 Tensor decompositions and N -way data arrays

Low-rank decompositions can be generalized beyond matrices. This is important as the SVD requires that disparate types of data be flattened into a single vector in order to evaluate correlated structures. For instance, different time snapshots (columns) of a matrix may include measurements as diverse as temperature, pressure, concentration of a substance, etc. Additionally, there may be categorical data. Vectorizing this data generally does not make sense. Ultimately, what is desired is to preserve the various data structures and types in

their own, independent directions. Matrices can be generalized to N -way arrays, or tensors, where the data is more appropriately arranged without forcing a data-flattening process.

The construction of data tensors requires that we revisit the notation associated with tensor addition, multiplication, and inner products [299]. We denote the r th column of a matrix \mathbf{A} by \mathbf{a}_r . Given matrices $\mathbf{A} \in \mathbb{R}^{I \times K}$ and $\mathbf{B} \in \mathbb{R}^{J \times K}$, their Khatri-Rao product is denoted by $\mathbf{A} \odot \mathbf{B}$ and is defined to be the $IJ \times K$ matrix of column-wise Kronecker products, namely

$$\mathbf{A} \odot \mathbf{B} = (\mathbf{a}_1 \otimes \mathbf{b}_1 \ \cdots \ \mathbf{a}_K \otimes \mathbf{b}_K).$$

For an N -way tensor \mathcal{A} of size $I_1 \times I_2 \times \cdots \times I_N$, we denote its $\mathbf{i} = (i_1, i_2, \dots, i_N)$ entry by $a_{\mathbf{i}}$.

The inner product between two N -way tensors \mathcal{A} and \mathcal{B} of compatible dimensions is given by

$$\langle \mathcal{A}, \mathcal{B} \rangle = \sum_{\mathbf{i}} a_{\mathbf{i}} b_{\mathbf{i}}.$$

The Frobenius norm of a tensor \mathcal{A} , denoted by $\|\mathcal{A}\|_F$, is the square root of the inner product of \mathcal{A} with itself, namely $\|\mathcal{A}\|_F = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$. Finally, the mode- n matricization or unfolding of a tensor \mathcal{A} is denoted by $\mathbf{mA}_{(n)}$.

Let \mathcal{M} represent an N -way data tensor of size $I_1 \times I_2 \times \cdots \times I_N$. We are interested in an R -component CANDECOMP/PARAFAC (CP) [124, 235, 299] factor model

$$\mathcal{M} = \sum_{r=1}^R \lambda_r \mathbf{m}\mathbf{a}_r^{(1)} \circ \cdots \circ \mathbf{m}\mathbf{a}_r^{(N)}, \quad (1.44)$$

where \circ represents outer product and $\mathbf{m}\mathbf{a}_r^{(n)}$ represents the r th column of the factor matrix $\mathbf{mA}^{(n)}$ of size $I_n \times R$. The CP decomposition refers to CANDECOMP/PARAFAC which stand for *parallel factors analysis* (PARAFAC) and *canonical decomposition* (CANDECOMP) respectively. We refer to each summand as a *component*. Assuming each factor matrix has been column-normalized to have unit Euclidean length, we refer to the λ_r 's as *weights*. We will use the shorthand notation where $\lambda = (\lambda_1, \dots, \lambda_R)^T$ [25]. A tensor that has a CP decomposition is sometimes referred to as a Kruskal tensor.

For the rest of this chapter, we consider a 3-way CP tensor decomposition (See Fig. 1.29) where two modes index state variation and the third mode indexes time variation:

$$\mathcal{M} = \sum_{r=1}^R \lambda_r \mathbf{A}_r \circ \mathbf{B}_r \circ \mathbf{C}_r.$$

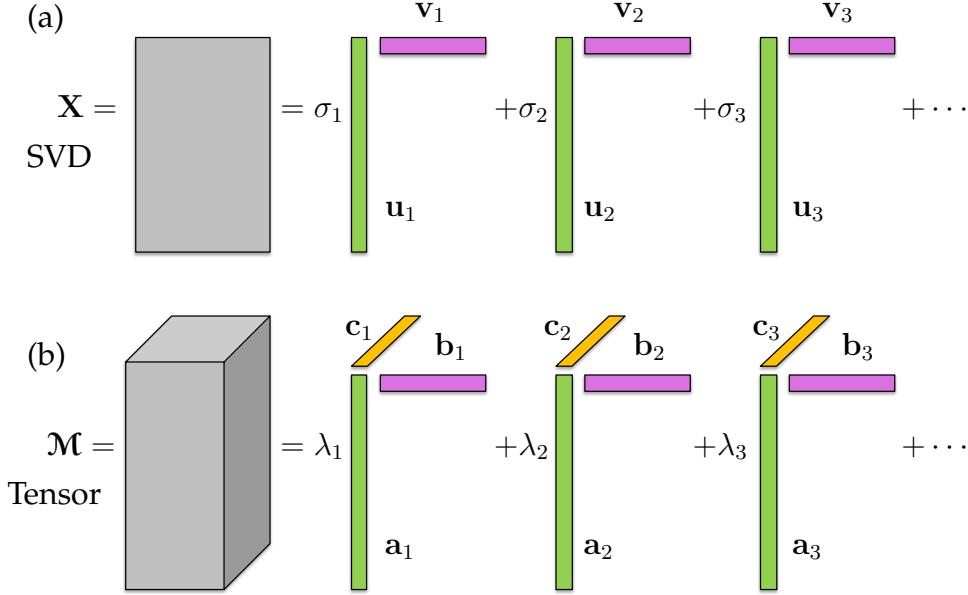


Figure 1.29: Comparison of the SVD and Tensor decomposition frameworks. Both methods produce an approximation to the original data matrix by sums of outer products. Specifically, the tensor decomposition generalizes the concept of the SVD to N -way arrays of data without having to flatten (vectorize) the data.

Let $A \in \mathbb{R}^{I_1 \times R}$ and $B \in \mathbb{R}^{I_2 \times R}$ denote the factor matrices corresponding to the two state modes and $C \in \mathbb{R}^{I_3 \times R}$ denote the factor matrix corresponding to the time mode. This 3-way decomposition is compared to the SVD in Fig. 1.29.

To illustrate the tensor decomposition, we use the MATLAB N -way toolbox developed by Rasmus Bro and co-workers [84, 15] which is available on the Mathworks file exchange. This simple to use package provides a variety of tools to extract tensor decompositions and evaluate the factor models generated. In the specific example considered here, we generate data from a spatio-temporal function (See Fig. 1.30)

$$F(x, y, t) = \exp(-x^2 - 0.5y^2) \cos(2t) + \operatorname{sech}(x) \tanh(x) \exp(-0.2y^2) \sin(t). \quad (1.45)$$

This model has two spatial modes with two distinct temporal frequencies, thus a two factor model should be sufficient to extract the underlying spatial and temporal modes. To construct this function in MATLAB, the following code is used.

Code 1.25: Creating tensor data.

```
|| x=-5:0.1:5; y=-6:0.1:6; t=0:0.1:10*pi;
```

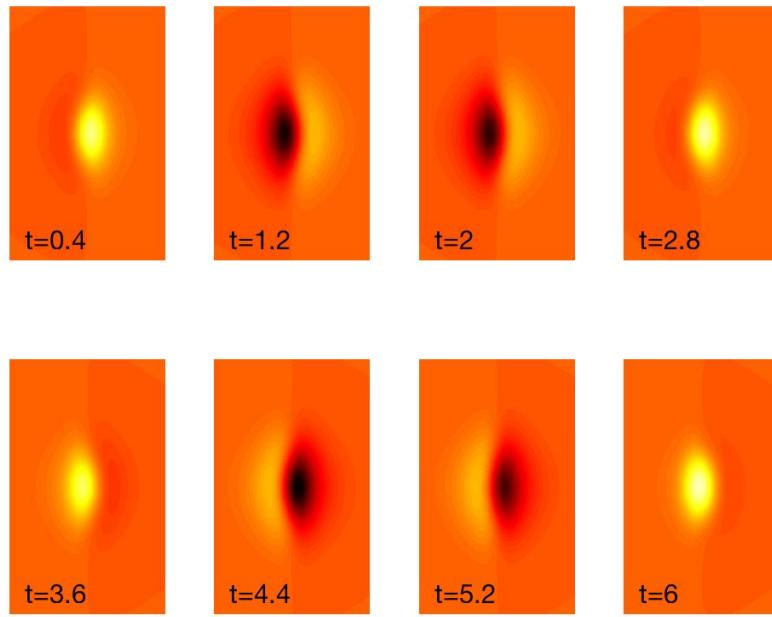


Figure 1.30: Example N -way array data set created from the function (1.45). The data matrix is $\mathbf{A} \in \mathbb{R}^{121 \times 101 \times 315}$. A CP tensor decomposition can be used to extract the two underlying structures that produced the data.

```

|| [X, Y, T] = meshgrid(x, y, t);
|| A = exp(-(X.^2 + 0.5*Y.^2)) .* (cos(2*T)) + ...
|| (sech(X) .* tanh(X) .* exp(-0.2*Y.^2)) .* sin(T);

```

Note that the **meshgrid** command is capable of generating N -way arrays. Indeed, MATLAB has no difficulties specifying higher-dimensional arrays and tensors. Specifically, one can easily generate N -way data matrices with arbitrary dimensions. The command $\mathbf{A} = \text{randn}(10, 10, 10, 10, 10)$ generates a 5-way hypercube with random values in each of the five directions of the array.

Figure 1.30 shows eight snapshots of the function (1.45) discretized with the code above. The N -way array data generated from the MATLAB code produces $\mathbf{A} \in \mathbb{R}^{121 \times 101 \times 315}$, which is of total dimension 10^6 . The CP tensor decomposition can be used to extract a two factor model for this 3-way array, thus producing two vectors in each direction of space x , space y , and time t .

The N -way toolbox provides a simple architecture for performing tensor decompositions. The PARAFAC command structure can easily take the input function (1.45) which is discretized in the code above and provide a two-factor model. The following code produces the output as model.

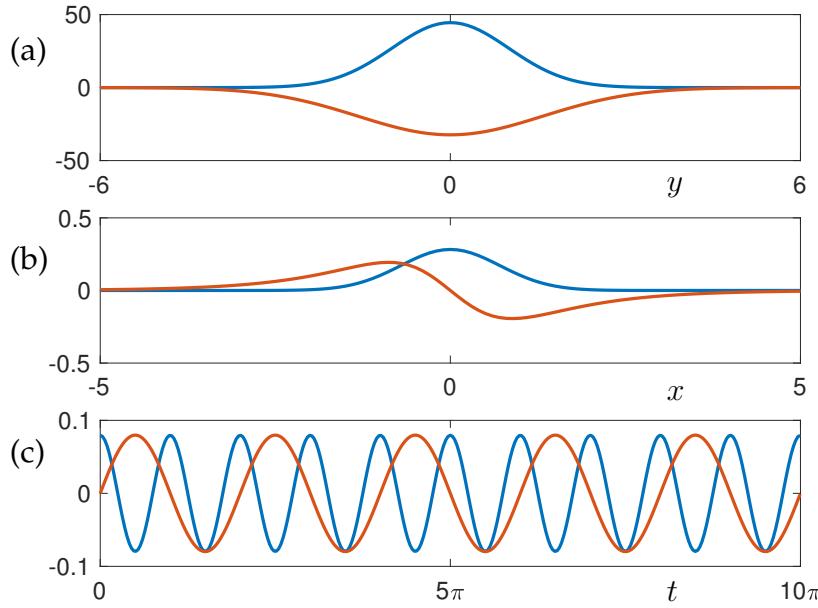


Figure 1.31: 3-way tensor decomposition of the function (1.45) discretized so that the data matrix is $\mathbf{A} \in \mathbb{R}^{121 \times 101 \times 315}$. A CP tensor decomposition can be used to extract the two underlying structures that produced the data. The first factor is in blue, the second factor is in red. The three distinct directions of the data (parallel factors) are illustrated in (a) the y direction, (b) the x direction, and (c) the time t .

Code 1.26: Two factor tensor model.

```
model=parafac(A, 2);
[A1,A2,A3]=fac2let(model);
subplot(3,1,1), plot(y,A1,'LineWidth',[2])
subplot(3,1,2), plot(x,A2,'LineWidth',[2])
subplot(3,1,3), plot(t,A3,'LineWidth',[2])
```

Note that in the above code, the **fac2let** command turns the factors in the model into their component matrices. Further note that the **meshgrid** arrangement of the data is different from **parafac** since the x and y directions are switched.

Figure 1.31 shows the results of the N -way tensor decomposition for the prescribed two factor model. Specifically, the two vectors along each of the three directions of the array are illustrated. For this example, the exact answer is known since the data was constructed from the rank-2 model (1.45). The first set of two modes (along the original y direction) are Gaussian as prescribed. The second set of two modes (along the original x direction) include a Gaussian for the first function, and the anti-symmetric $\text{sech}(x) \tanh(x)$ for the second function. The third set of two modes correspond to the time dynamics of the two functions: $\cos(2t)$ and $\sin(t)$, respectively. Thus, the two factor model

produced by the CP tensor decomposition returns the expected, low-rank functions that produced the high-dimensional data matrix \mathbf{A} .

Recent theoretical and computational advances in N -way decompositions are opening up the potential for tensor decompositions in many fields. For N large, such decompositions can be computationally intractable due to the size of the data. Indeed, even in the simple example illustrated in Figs. 1.30 and 1.31, there are 10^6 data points. Ultimately, the CP tensor decomposition does not scale well with additional data dimensions. However, randomized techniques are helping yield tractable computations even for large data sets [158, 175]. As with the SVD, randomized methods exploit the underlying low-rank structure of the data in order to produce an accurate approximation through the sum of rank-one outer products. Additionally, tensor decompositions can be combined with constraints on the form of the parallel factors in order to produce more easily interpretable results [348]. This gives a framework for producing interpretable and scalable computations of N -way data arrays.

Suggested reading

Texts

- (1) **Matrix computations**, by G. H. Golub and C. F. Van Loan, 2012 [214].

Papers and reviews

- (1) **Calculating the singular values and pseudo-inverse of a matrix**, by G. H. Golub and W. Kahan, *Journal of the Society for Industrial & Applied Mathematics, Series B: Numerical Analysis*, 1965 [211].
- (2) **A low-dimensional procedure for the characterization of human faces**, by L. Sirovich and M. Kirby, *Journal of the Optical Society of America A*, 1987 [491].
- (3) **Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions**, by N. Halko, P.-G. Martinsson, and J. A. Tropp, *SIAM Review*, 2011 [230].
- (4) **A randomized algorithm for the decomposition of matrices**, by P.-G. Martinsson, V. Rokhlin, and M. Tygert, *Applied and Computational Harmonic Analysis*, 2011 [371].
- (5) **The optimal hard threshold for singular values is $4/\sqrt{3}$** , by M. Gavish and D. L. Donoho, *IEEE Transactions on Information Theory*, 2014 [200].

Bibliography

- [1] R. Abraham and J. E. Marsden. *Foundations of mechanics*, volume 36. Benjamin/Cummings Publishing Company Reading, Massachusetts, 1978.
- [2] R. Abraham, J. E. Marsden, and T. Ratiu. *Manifolds, Tensor Analysis, and Applications*, volume 75 of *Applied Mathematical Sciences*. Springer-Verlag, 1988.
- [3] M. Agrawal, S. Vidyashankar, and K. Huang. On-chip implementation of ECoG signal data decoding in brain-computer interface. In *Mixed-Signal Testing Workshop (IMSTW), 2016 IEEE 21st International*, pages 1–6. IEEE, 2016.
- [4] R. Agrawal, R. Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB*, volume 1215, pages 487–499, 1994.
- [5] H.-S. Ahn, Y. Chen, and K. L. Moore. Iterative learning control: Brief survey and categorization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1099–1121, 2007.
- [6] H. Akaike. Fitting autoregressive models for prediction. *Annals of the institute of Statistical Mathematics*, 21(1):243–247, 1969.
- [7] H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, 1974.
- [8] W. Amrein and A.-M. Berthier. On support properties of L^p -functions and their Fourier transforms. *Journal of Functional Analysis*, 24(3):258–267, 1977.
- [9] D. Amsallem, J. Cortial, and C. Farhat. On-demand cfd-based aeroelastic predictions using a database of reduced-order bases and models. In *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, page 800, 2009.
- [10] D. Amsallem and C. Farhat. An online method for interpolating linear parametric reduced-order models. *SIAM Journal on Scientific Computing*, 33(5):2169–2198, 2011.
- [11] D. Amsallem, M. J. Zahr, and K. Washabaugh. Fast local reduced basis updates for the efficient reduction of nonlinear systems with hyper-reduction. *Advances in Computational Mathematics*, 41(5):1187–1230, 2015.
- [12] J. Andén and S. Mallat. Deep scattering spectrum. *IEEE Transactions on Signal Processing*, 62(16):4114–4128, 2014.
- [13] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammerling, A. McKenney, et al. *LAPACK Users' guide*, volume 9. Siam, 1999.
- [14] J. L. Anderson. An ensemble adjustment Kalman filter for data assimilation. *Monthly weather review*, 129(12):2884–2903, 2001.
- [15] C. A. Andersson and R. Bro. The n-way toolbox for matlab. *Chemometrics and intelligent laboratory systems*, 52(1):1–4, 2000.

- [16] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *IEEE Transactions on image processing*, 1(2):205–220, 1992.
- [17] A. C. Antoulas. *Approximation of large-scale dynamical systems*. SIAM, 2005.
- [18] H. Arbabi and I. Mezić. Ergodic theory, dynamic mode decomposition and computation of spectral properties of the Koopman operator. *SIAM J. Appl. Dyn. Syst.*, 16(4):2096–2126, 2017.
- [19] K. B. Ariyur and M. Krstić. *Real-Time Optimization by Extremum-Seeking Control*. Wiley, Hoboken, New Jersey, 2003.
- [20] T. Askham and J. N. Kutz. Variable projection methods for an optimized dynamic mode decomposition. *SIAM J. Appl. Dyn. Syst.*, 17(1):380–416, 2018.
- [21] P. Astrid. Fast reduced order modeling technique for large scale LTV systems. In *American Control Conference, 2004. Proceedings of the 2004*, volume 1, pages 762–767. IEEE, 2004.
- [22] K. J. Åström and R. M. Murray. *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2010.
- [23] M. Azeez and A. Vakakis. Proper orthogonal decomposition (POD) of a class of vibroimpact oscillations. *Journal of Sound and vibration*, 240(5):859–889, 2001.
- [24] K. Bache and M. Lichman. UCI machine learning repository, 2013.
- [25] B. W. Bader and T. G. Kolda. Efficient MATLAB computations with sparse and factored tensors. *SIAM Journal on Scientific Computing*, 30(1):205–231, Dec. 2007.
- [26] S. Bagheri. Koopman-mode decomposition of the cylinder wake. *Journal of Fluid Mechanics*, 726:596–623, 2013.
- [27] S. Bagheri, L. Brandt, and D. Henningson. Input-output analysis, model reduction and control of the flat-plate boundary layer. *J. Fluid Mechanics*, 620:263–298, 2009.
- [28] S. Bagheri, J. Hoepffner, P. J. Schmid, and D. S. Henningson. Input-output analysis and control design applied to a linear model of spatially developing flows. *Appl. Mech. Rev.*, 62(2):020803–1..27, 2009.
- [29] Z. Bai, S. L. Brunton, B. W. Brunton, J. N. Kutz, E. Kaiser, A. Spohn, and B. R. Noack. Data-driven methods in fluid dynamics: Sparse classification from experimental data. In *invited chapter for Whither Turbulence and Big Data in the 21st Century*, 2015.
- [30] Z. Bai, E. Kaiser, J. L. Proctor, J. N. Kutz, and S. L. Brunton. Dynamic mode decomposition for compressive system identification. *arXiv preprint arXiv:1710.07737*, 2017.
- [31] Z. Bai, T. Wimalajeewa, Z. Berger, G. Wang, M. Glauser, and P. K. Varshney. Low-dimensional approach for reconstruction of airfoil data via compressive sensing. *AIAA Journal*, 53(4):920–933, 2014.
- [32] M. J. Balajewicz, E. H. Dowell, and B. R. Noack. Low-dimensional modelling of high-Reynolds-number shear flows incorporating constraints from the Navier–Stokes equation. *Journal of Fluid Mechanics*, 729:285–308, 2013.
- [33] M. Balasubramanian, S. Zabic, C. Bowd, H. W. Thompson, P. Wolenski, S. S. Iyengar, B. B. Karki, and L. M. Zangwill. A framework for detecting glaucomatous progression in the optic nerve head of an eye using proper orthogonal decomposition. *IEEE Transactions on Information Technology in Biomedicine*, 13(5):781–793, 2009.
- [34] B. Bamieh and L. Giarré. Identification of linear parameter varying models. *International Journal of Robust and Nonlinear Control*, 12:841–853, 2002.

- [35] A. Banaszuk, K. B. Ariyur, M. Krstić, and C. A. Jacobson. An adaptive algorithm for control of combustion instability. *Automatica*, 40(11):1965–1972, 2004.
- [36] A. Banaszuk, S. Narayanan, and Y. Zhang. Adaptive control of flow separation in a planar diffuser. *AIAA paper*, 617:2003, 2003.
- [37] A. Banaszuk, Y. Zhang, and C. A. Jacobson. Adaptive control of combustion instability using extremum-seeking. In *American Control Conference, 2000. Proceedings of the 2000*, volume 1, pages 416–422. IEEE, 2000.
- [38] S. Banks. Infinite-dimensional Carleman linearization, the Lie series and optimal control of non-linear partial differential equations. *International journal of systems science*, 23(5):663–675, 1992.
- [39] R. G. Baraniuk. Compressive sensing. *IEEE Signal Processing Magazine*, 24(4):118–120, 2007.
- [40] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56(4):1982–2001, 2010.
- [41] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An empirical interpolation method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathematique*, 339(9):667–672, 2004.
- [42] J. Basley, L. R. Pastur, N. Delprat, and F. Lusseyran. Space-time aspects of a three-dimensional multi-modulated open cavity flow. *Physics of Fluids (1994-present)*, 25(6):064105, 2013.
- [43] J. Basley, L. R. Pastur, F. Lusseyran, T. M. Faure, and N. Delprat. Experimental investigation of global structures in an incompressible cavity flow using time-resolved PIV. *Experiments in Fluids*, 50(4):905–918, 2011.
- [44] W. Baur and V. Strassen. The complexity of partial derivatives. *Theoretical computer science*, 22(3):317–330, 1983.
- [45] J. F. Beaudoin, O. Cadot, J. L. Aider, and J. E. Wesfreid. Bluff-body drag reduction by extremum-seeking control. *Journal of Fluids and Structures*, 22:973–978, 2006.
- [46] J.-F. Beaudoin, O. Cadot, J.-L. Aider, and J.-E. Wesfreid. Drag reduction of a bluff body using adaptive control methods. *Physics of Fluids*, 18(8):085107, 2006.
- [47] R. Becker, R. King, R. Petz, and W. Nitsche. Adaptive closed-loop control on a high-lift configuration using extremum seeking. *AIAA Journal*, 45(6):1382–92, 2007.
- [48] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):711–720, 1997.
- [49] G. Bellani. Experimental studies of complex flows through image-based techniques. 2011.
- [50] B. A. Belson, J. H. Tu, and C. W. Rowley. Algorithm 945: modred—a parallelized model reduction library. *ACM Transactions on Mathematical Software*, 40(4):30, 2014.
- [51] M. Benedicks. On Fourier transforms of functions supported on sets of finite Lebesgue measure. *Journal of mathematical analysis and applications*, 106(1):180–183, 1985.
- [52] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160, 2007.

- [53] P. Benner, A. Cohen, M. Ohlberger, and K. Willcox. *Model Reduction and Approximation: Theory and Algorithms*, volume 15. SIAM, 2017.
- [54] P. Benner, S. Gugercin, and K. Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev.*, 57(4):483–531, 2015.
- [55] P. Benner, J.-R. Li, and T. Penzl. Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems. *Numerical Linear Algebra with Applications*, 15(9):755–777, 2008.
- [56] E. Berger, M. Sastuba, D. Vogt, B. Jung, and H. B. Amor. Estimation of perturbations in robotic behavior using dynamic mode decomposition. *Journal of Advanced Robotics*, 29(5):331–343, 2015.
- [57] G. Berkooz, P. Holmes, and J. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Ann. Rev. Fluid Mech.*, 25:539–575, 1993.
- [58] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms i. *Communications on pure and applied mathematics*, 44(2):141–183, 1991.
- [59] S. A. Billings. *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons, 2013.
- [60] P. Binetti, K. B. Ariyur, M. Krstić, and F. Bernelli. Formation flight optimization using extremum seeking feedback. *Journal of Guidance, Control, and Dynamics*, 26(1):132–142, 2003.
- [61] G. D. Birkhoff. Proof of the ergodic theorem. *Proceedings of the National Academy of Sciences*, 17(12):656–660, 1931.
- [62] G. D. Birkhoff and B. O. Koopman. Recent contributions to the ergodic theory. *Proceedings of the National Academy of Sciences*, 18(3):279–282, 1932.
- [63] C. M. Bishop. *Neural networks for pattern recognition*. Oxford university press, 1995.
- [64] C. M. Bishop. *Pattern recognition and machine learning*. Springer New York, 2006.
- [65] D. Bistrian and I. Navon. Randomized dynamic mode decomposition for non-intrusive reduced order modelling. *International Journal for Numerical Methods in Engineering*, 2016.
- [66] D. A. Bistrian and I. M. Navon. An improved algorithm for the shallow water equations model reduction: Dynamic mode decomposition vs POD. *International Journal for Numerical Methods in Fluids*, 2015.
- [67] P. Bondi, G. Casalino, and L. Gambardella. On the iterative learning control theory for robotic manipulators. *IEEE Journal on Robotics and Automation*, 4(1):14–22, 1988.
- [68] J. Bongard and H. Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007.
- [69] J. L. Borges. The library of Babel. *Collected fictions*, 1998.
- [70] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.
- [71] H. Boulard and Y. Kamp. Autoassociative memory by multilayer perceptron and singular values decomposition. *Biol Cybern*, 59:291–294, 1989.
- [72] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

- [73] S. Boyd, L. O. Chua, and C. A. Desoer. Analytical foundations of volterra series. *IMA Journal of Mathematical Control and Information*, 1(3):243–282, 1984.
- [74] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2009.
- [75] A. I. Bratcu, I. Munteanu, S. Bacha, and B. Raison. Maximum power point tracking of grid-connected photovoltaic arrays by using extremum seeking control. *CEAI*, 10(4):3–12, 2008.
- [76] L. Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, 1995.
- [77] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [78] L. Breiman et al. Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231, 2001.
- [79] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.
- [80] I. Bright, G. Lin, and J. N. Kutz. Compressive sensing and machine learning strategies for characterizing the flow around a cylinder with limited pressure measurements. *Physics of Fluids*, 25(127102):1–15, 2013.
- [81] I. Bright, G. Lin, and J. N. Kutz. Classification of spatio-temporal data via asynchronous sparse sampling: Application to flow around a cylinder. *SIAM Multiscale modeling and simulation*, 14(2):823–838, 2016.
- [82] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.
- [83] D. Bristow, M. Tharayil, A. G. Alleyne, et al. A survey of iterative learning control. *Control Systems, IEEE*, 26(3):96–114, 2006.
- [84] R. Bro. Parafac. tutorial and applications. *Chemometrics and intelligent laboratory systems*, 38(2):149–171, 1997.
- [85] A. Broad, T. Murphrey, and B. Argall. Learning models for shared control of human-machine systems with unknown dynamics. *Robotics: Science and Systems Proceedings*, 2017.
- [86] R. W. Brockett. Volterra series and geometric control theory. *Automatica*, 12(2):167–176, 1976.
- [87] D. Broomhead and R. Jones. Time-series analysis. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 423, pages 103–121. The Royal Society, 1989.
- [88] D. S. Broomhead and D. Lowe. Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, Royal Signals and Radar Establishment Malvern (United Kingdom), 1988.
- [89] B. W. Brunton, S. L. Brunton, J. L. Proctor, and J. N. Kutz. Sparse sensor placement optimization for classification. *SIAM Journal on Applied Mathematics*, 76(5):2099–2122, 2016.
- [90] B. W. Brunton, L. A. Johnson, J. G. Ojemann, and J. N. Kutz. Extracting spatial-temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition. *Journal of Neuroscience Methods*, 258:1–15, 2016.

- [91] S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz. Chaos as an intermittently forced linear system. *Nature Communications*, 8(19):1–9, 2017.
- [92] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PLoS ONE*, 11(2):e0150171, 2016.
- [93] S. L. Brunton, X. Fu, and J. N. Kutz. Extremum-seeking control of a mode-locked laser. *IEEE Journal of Quantum Electronics*, 49(10):852–861, 2013.
- [94] S. L. Brunton, X. Fu, and J. N. Kutz. Self-tuning fiber lasers. *IEEE Journal of Selected Topics in Quantum Electronics*, 20(5), 2014.
- [95] S. L. Brunton and B. R. Noack. Closed-loop turbulence control: Progress and challenges. *Applied Mechanics Reviews*, 67:050801–1–050801–48, 2015.
- [96] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.
- [97] S. L. Brunton, J. L. Proctor, and J. N. Kutz. Sparse identification of nonlinear dynamics with control (SINDYC). *IFAC NOLCOS*, 49(18):710–715, 2016.
- [98] S. L. Brunton, J. L. Proctor, J. H. Tu, and J. N. Kutz. Compressed sensing and dynamic mode decomposition. *Journal of Computational Dynamics*, 2(2):165–191, 2015.
- [99] S. L. Brunton and C. W. Rowley. Maximum power point tracking for photovoltaic optimization using ripple-based extremum seeking control. *IEEE Transactions on Power Electronics*, 25(10):2531–2540, 2010.
- [100] S. L. Brunton, J. H. Tu, I. Bright, and J. N. Kutz. Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems. *SIAM Journal on Applied Dynamical Systems*, 13(4):1716–1732, 2014.
- [101] D. Buche, P. Stoll, R. Dornberger, and P. Koumoutsakos. Multiobjective evolutionary algorithm for the optimization of noisy combustion processes. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 32(4):460–473, 2002.
- [102] M. Budišić and I. Mezić. An approximate parametrization of the ergodic partition using time averaged observables. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 3162–3168. IEEE, 2009.
- [103] M. Budišić and I. Mezić. Geometry of the ergodic quotient reveals coherent structures in flows. *Physica D: Nonlinear Phenomena*, 241(15):1255–1269, 2012.
- [104] M. Budišić, R. Mohr, and I. Mezić. Applied Koopmanism a). *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4):047510, 2012.
- [105] K. P. Burnham and D. R. Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer Science & Business Media, 2003.
- [106] P. A. Businger and G. H. Golub. Algorithm 358: Singular value decomposition of a complex matrix [f1, 4, 5]. *Communications of the ACM*, 12(10):564–565, 1969.
- [107] E. F. Camacho and C. B. Alba. *Model predictive control*. Springer Science & Business Media, 2013.

- [108] E. Cambria, G.-B. Huang, L. L. C. Kasun, H. Zhou, C. M. Vong, J. Lin, J. Yin, Z. Cai, Q. Liu, K. Li, et al. Extreme learning machines [trends & controversies]. *IEEE Intelligent Systems*, 28(6):30–59, 2013.
- [109] E. J. Candès. Compressive sensing. *Proceedings of the International Congress of Mathematics*, 2006.
- [110] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):11–11–37, 2011.
- [111] E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.
- [112] E. J. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications in Pure and Applied Mathematics*, 8(1207–1223), 59.
- [113] E. J. Candès and T. Tao. Decoding by linear programming. *Information Theory, IEEE Transactions on*, 51(12):4203–4215, 2005.
- [114] E. J. Candès and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12):5406–5425, 2006.
- [115] E. J. Candès and M. B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, pages 21–30, 2008.
- [116] Y. Cao, J. Zhu, Z. Luo, and I. Navon. Reduced-order modeling of the upper tropical pacific ocean model using proper orthogonal decomposition. *Computers & Mathematics with Applications*, 52(8):1373–1386, 2006.
- [117] Y. Cao, J. Zhu, I. M. Navon, and Z. Luo. A reduced-order approach to four-dimensional variational data assimilation using proper orthogonal decomposition. *International Journal for Numerical Methods in Fluids*, 53(10):1571–1583, 2007.
- [118] K. Carlberg, M. Barone, and H. Antil. Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction. *Journal of Computational Physics*, 330:693–734, 2017.
- [119] K. Carlberg, C. Bou-Mosleh, and C. Farhat. Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximations. *International Journal for Numerical Methods in Engineering*, 86(2):155–181, 2011.
- [120] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem. The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics*, 242:623–647, 2013.
- [121] T. Carleman. Application de la théorie des équations intégrales linéaires aux systèmes d'équations différentielles non linéaires. *Acta Mathematica*, 59(1):63–87, 1932.
- [122] T. Carleman. Sur la théorie de l'équation intégrodifférentielle de boltzmann. *Acta Mathematica*, 60(1):91–146, 1933.
- [123] T. Carleman. Sur les systemes lineaires aux dérivées partielles du premier ordre a deux variables. *CR Acad. Sci. Paris*, 197:471–474, 1933.
- [124] J. D. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of “Eckart-Young” decomposition. *Psychometrika*, 35:283–319, 1970.

- [125] R. Chartrand. Numerical differentiation of noisy, nonsmooth data. *ISRN Applied Mathematics*, 2011, 2011.
- [126] A. Chatterjee. An introduction to the proper orthogonal decomposition. *Current science*, 78(7):808–817, 2000.
- [127] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.
- [128] K. K. Chen and C. W. Rowley. Normalized coprime robust stability and performance guarantees for reduced-order controllers. *IEEE Transactions on Automatic Control*, 58(4):1068–1073, 2013.
- [129] K. K. Chen, J. H. Tu, and C. W. Rowley. Variants of dynamic mode decomposition: Boundary condition, Koopman, and Fourier analyses. *Journal of Nonlinear Science*, 22(6):887–915, 2012.
- [130] Y. Chen, K. L. Moore, and H.-S. Ahn. Iterative learning control. In *Encyclopedia of the Sciences of Learning*, pages 1648–1652. Springer, 2012.
- [131] S. Cherry. Singular value decomposition analysis and canonical correlation analysis. *Journal of Climate*, 9(9):2003–2009, 1996.
- [132] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [133] J. Choi, M. Krstić, K. Ariyur, and J. Lee. Extremum seeking control for discrete-time systems. *IEEE Transactions on Automatic Control*, 47(2):318–323, FEB 2002.
- [134] Y. Choi, D. Amsalem, and C. Farhat. Gradient-based constrained optimization using a database of linear reduced-order models. *arXiv preprint arXiv:1506.07849*, 2015.
- [135] T. Colonius and K. Taira. A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 197:2131–2146, 2008.
- [136] J. W. Cooley, P. A. Lewis, and P. D. Welch. Historical notes on the fast Fourier transform. *Proceedings of the IEEE*, 55(10):1675–1677, 1967.
- [137] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation*, 19(90):297–301, 1965.
- [138] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [139] M. C. Cross and P. C. Hohenberg. Pattern formation outside of equilibrium. *Reviews of modern physics*, 65(3):851, 1993.
- [140] J. P. Crutchfield and B. S. McNamara. Equations of motion from a data series. *Complex systems*, 1:417–452, 1987.
- [141] M. Dam, M. Brøns, J. Juul Rasmussen, V. Naulin, and J. S. Hesthaven. Sparse identification of a predator-prey system from simulation data of a convection model. *Physics of Plasmas*, 24(2):022310, 2017.
- [142] B. C. Daniels and I. Nemenman. Automated adaptive inference of phenomenological dynamical models. *Nature communications*, 6, 2015.
- [143] B. C. Daniels and I. Nemenman. Efficient inference of parsimonious phenomenological models of cellular dynamics using s-systems and alternating regression. *PloS one*, 10(3):e0119821, 2015.

- [144] S. Das and D. Giannakis. Delay-coordinate maps and the spectra of Koopman operators. *arXiv preprint arXiv:1706.08544*, 2017.
- [145] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE transactions on information theory*, 36(5):961–1005, 1990.
- [146] L. Davis et al. *Handbook of genetic algorithms*, volume 115. Van Nostrand Reinhold New York, 1991.
- [147] S. T. Dawson, M. S. Hemati, M. O. Williams, and C. W. Rowley. Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. *Experiments in Fluids*, 57(3):1–19, 2016.
- [148] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.
- [149] S. Devasia, D. Chen, and B. Paden. Nonlinear inversion-based output tracking. *Automatic Control, IEEE Transactions on*, 41(7):930–942, 1996.
- [150] D. Donoho. 50 years of data science. In *Based on a Presentation at the Tukey Centennial Workshop*. NJ Princeton, 2015.
- [151] D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.
- [152] D. L. Donoho and M. Gavish. Code supplement to "The optimal hard threshold for singular values is $4/\sqrt{3}$ ". <http://purl.stanford.edu/vg705qn9070>, 2014.
- [153] D. L. Donoho, I. M. Johnstone, J. C. Hoch, and A. S. Stern. Maximum entropy and the nearly black object. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 41–81, 1992.
- [154] D. L. Donoho and J. M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.
- [155] J. C. Doyle. Guaranteed margins for LQG regulators. *IEEE Transactions on Automatic Control*, 23(4):756–757, 1978.
- [156] J. C. Doyle, B. A. Francis, and A. R. Tannenbaum. *Feedback control theory*. Courier Corporation, 2013.
- [157] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis. State-space solutions to standard H_2 and H_∞ control problems. *IEEE Transactions on Automatic Control*, 34(8):831–847, 1989.
- [158] P. Drineas and M. W. Mahoney. A randomized algorithm for a tensor-based generalization of the singular value decomposition. *Linear algebra and its applications*, 420(2-3):553–571, 2007.
- [159] Z. Drmac and S. Gugercin. A new selection operator for the discrete empirical interpolation method—improved a priori error bound and extensions. *SIAM Journal on Scientific Computing*, 38(2):A631–A648, 2016.
- [160] Q. Du and M. Gunzburger. Model reduction by proper orthogonal decomposition coupled with centroidal voronoi tessellations (keynote). In *ASME 2002 Joint US-European Fluids Engineering Division Conference*, pages 1401–1406. American Society of Mechanical Engineers, 2002.
- [161] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

- [162] J. A. Duersch and M. Gu. Randomized QR with column pivoting. *SIAM Journal on Scientific Computing*, 39(4):C263–C291, 2017.
- [163] D. Duke, D. Honnery, and J. Soria. Experimental investigation of nonlinear instabilities in annular liquid sheets. *Journal of Fluid Mechanics*, 691:594–604, 2012.
- [164] D. Duke, J. Soria, and D. Honnery. An error analysis of the dynamic mode decomposition. *Experiments in fluids*, 52(2):529–542, 2012.
- [165] G. E. Dullerud and F. Paganini. *A course in robust control theory: A convex approach*. Texts in Applied Mathematics. Springer, Berlin, Heidelberg, 2000.
- [166] R. Dunne and B. J. McKeon. Dynamic stall on a pitching and surging airfoil. *Experiments in Fluids*, 56(8):1–15, 2015.
- [167] T. Duriez, S. L. Brunton, and B. R. Noack. *Machine Learning Control: Taming Nonlinear Dynamics and Turbulence*. Springer, 2016.
- [168] T. Duriez, V. Parezanović, L. Cordier, B. R. Noack, J. Delville, J.-P. Bonnet, M. Segond, and M. Abel. Closed-loop turbulence control using machine learning. *arXiv preprint arXiv:1404.4589*, 2014.
- [169] T. Duriez, V. Parezanovic, J.-C. Laurentie, C. Fourment, J. Delville, J.-P. Bonnet, L. Cordier, B. R. Noack, M. Segond, M. Abel, N. Gautier, J.-L. Aider, C. Raibaud, C. Cuvier, M. Stanislas, and S. L. Brunton. Closed-loop control of experimental shear flows using machine learning. AIAA Paper 2014-2219, 7th Flow Control Conference, 2014.
- [170] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.
- [171] J. L. Eftang, A. T. Patera, and E. M. Rønquist. An “hp” certified reduced basis method for parametrized elliptic partial differential equations. *SIAM Journal on Scientific Computing*, 32(6):3170–3200, 2010.
- [172] J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.
- [173] U. Eren, A. Prach, B. B. Koçer, S. V. Raković, E. Kayacan, and B. Açıkmese. Model predictive control in aerospace systems: Current state and opportunities. *Journal of Guidance, Control, and Dynamics*, 40(7):1541–1566, 2017.
- [174] N. B. Erichson, S. L. Brunton, and J. N. Kutz. Compressed dynamic mode decomposition for real-time object detection. *Journal of Real-Time Image Processing*, 2016.
- [175] N. B. Erichson, S. L. Brunton, and J. N. Kutz. Randomized dynamic mode decomposition. *arXiv preprint arXiv:1702.02912*, 2017.
- [176] N. B. Erichson, K. Manohar, S. L. Brunton, and J. N. Kutz. Randomized CP tensor decomposition. *arXiv preprint arXiv:1703.09074*.
- [177] N. B. Erichson, S. Voronin, S. L. Brunton, and J. N. Kutz. Randomized matrix decompositions using R. *arXiv preprint arXiv:1608.02148*, 2016.
- [178] T. Esram, J. W. Kimball, P. T Krein, P. L. Chapman, and P. Midya. Dynamic maximum power point tracking of photovoltaic arrays using ripple correlation control. *Ieee Transactions On Power Electronics*, 21(5):1282–1291, Sept. 2006.
- [179] R. Everson and L. Sirovich. Karhunen–Loeve procedure for gappy data. *JOSA A*, 12(8):1657–1664, 1995.

- [180] N. Fabbiane, O. Semeraro, S. Bagheri, and D. S. Henningson. Adaptive and model-based control theory applied to convectively unstable flows. *Appl. Mech. Rev.*, 66(6):060801–1–20, 2014.
- [181] B. Feeny. On proper orthogonal co-ordinates as indicators of modal activity. *Journal of Sound and Vibration*, 255(5):805–817, 2002.
- [182] R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 222:309–368, 1922.
- [183] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of human genetics*, 7(2):179–188, 1936.
- [184] P. J. Fleming and R. C. Purshouse. Evolutionary algorithms in control systems engineering: a survey. *Control Engineering Practice*, 10:1223–1241, 2002.
- [185] J. Fourier. *Theorie analytique de la chaleur, par M. Fourier*. Chez Firmin Didot, père et fils, 1822.
- [186] J. B. J. Fourier. *The analytical theory of heat*. The University Press, 1878.
- [187] J. E. Fowler. Compressive-projection principal component analysis. *IEEE Transactions on Image Processing*, 18(10):2230–2242, 2009.
- [188] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.
- [189] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.
- [190] A. Frieze, R. Kannan, and S. Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51(6):1025–1041, 2004.
- [191] X. Fu, S. L. Brunton, and J. Nathan Kutz. Classification of birefringence in mode-locked fiber lasers using machine learning and sparse representation. *Optics express*, 22(7):8585–8597, 2014.
- [192] K. Fukagata, S. Kern, P. Chatelain, P. Koumoutsakos, and N. Kasagi. Evolutionary optimization of an anisotropic compliant surface for turbulent friction drag reduction. *Journal of Turbulence*, 9(35):1–17, 2008.
- [193] F. Fukushima. A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetic*, 36:193–202, 1980.
- [194] H. Gao, J. Lam, C. Wang, and Y. Wang. Delay-dependent output-feedback stabilisation of discrete-time systems with time-varying state delay. *IEE Proceedings-Control Theory and Applications*, 151(6):691–698, 2004.
- [195] C. E. Garcia, D. M. Prett, and M. Morari. Model predictive control: theory and practice—a survey. *Automatica*, 25(3):335–348, 1989.
- [196] J. L. Garriga and M. Soroush. Model predictive control tuning methods: A review. *Industrial & Engineering Chemistry Research*, 49(8):3505–3515, 2010.
- [197] C. Gauss. Nachlass: Theoria interpolationis methodo nova tractata, volume werke. *Königliche Gesellschaft der Wissenschaften, Göttingen*, 1866.
- [198] C.-F. Gauss. *Theoria combinationis observationum erroribus minimis obnoxiae*, volume 1. Henricus Dieterich, 1823.

- [199] N. Gautier, J.-L. Aider, T. Duriez, B. Noack, M. Segond, and M. Abel. Closed-loop separation control using machine learning. *Journal of Fluid Mechanics*, 770:442–457, 2015.
- [200] M. Gavish and D. L. Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, 60(8):5040–5053, 2014.
- [201] M. Gazzola, O. V. Vasilyev, and P. Koumoutsakos. Shape optimization for drag reduction in linked bodies using evolution strategies. *Computers & Structures*, 89(11):1224–1231, 2011.
- [202] G. Gelbert, J. P. Moeck, C. O. Paschereit, and R. King. Advanced algorithms for gradient estimation in one-and two-parameter extremum seeking controllers. *Journal of Process Control*, 22(4):700–709, 2012.
- [203] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 23(6):643–660, 2001.
- [204] J. J. Gerbrands. On the relationships between SVD, KLT and PCA. *Pattern recognition*, 14(1):375–381, 1981.
- [205] A. C. Gilbert and P. Indyk. Sparse recovery using sparse matrices. *Proceedings of the IEEE*, 98(6):937–947, 2010.
- [206] A. C. Gilbert, J. Y. Park, and M. B. Wakin. Sketched SVD: Recovering spectral features from compressive measurements. *ArXiv e-prints*, 2012.
- [207] A. C. Gilbert, M. J. Strauss, and J. A. Tropp. A tutorial on fast Fourier sampling. *IEEE Signal Processing Magazine*, pages 57–66, 2008.
- [208] B. Glaz, L. Liu, and P. P. Friedmann. Reduced-order nonlinear unsteady aerodynamic modeling using a surrogate-based recurrence framework. *AIAA journal*, 48(10):2418–2429, 2010.
- [209] P. J. Goddard and K. Glover. Controller approximation: approaches for preserving H_∞ performance. *IEEE Transactions on Automatic Control*, 43(7):858–871, 1998.
- [210] D. E. Goldberg. *Genetic algorithms*. Pearson Education India, 2006.
- [211] G. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial & Applied Mathematics, Series B: Numerical Analysis*, 2(2):205–224, 1965.
- [212] G. Golub, S. Nash, and C. Van Loan. A Hessenberg-Schur method for the problem $ax + xb = c$. *IEEE Transactions on Automatic Control*, 24(6):909–913, 1979.
- [213] G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. *Numerical Mathematics*, 14:403–420, 1970.
- [214] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.
- [215] R. Gonzalez-Garcia, R. Rico-Martinez, and I. Kevrekidis. Identification of distributed parameter systems: A neural net based approach. *Comp. & Chem. Eng.*, 22:S965–S968, 1998.
- [216] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [217] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

- [218] M. Grant, S. Boyd, and Y. Ye. Cvx: Matlab software for disciplined convex programming, 2008.
- [219] A. Graves, G. Wayne, and I. Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.
- [220] A. Greenbaum. *Iterative methods for solving linear systems*. SIAM, 1997.
- [221] M. S. Grewal. Kalman filtering. In *International Encyclopedia of Statistical Science*, pages 705–708. Springer, 2011.
- [222] M. Grilli, P. J. Schmid, S. Hickel, and N. A. Adams. Analysis of unsteady behaviour in shockwave turbulent boundary layer interaction. *Journal of Fluid Mechanics*, 700:16–28, 2012.
- [223] J. Grosek and J. N. Kutz. Dynamic mode decomposition for real-time background/foreground separation in video. *arXiv preprint arXiv:1404.7592*, 2014.
- [224] M. Gu. Subspace iteration randomization and singular value problems. *SIAM Journal on Scientific Computing*, 37(3):1139–1173, 2015.
- [225] F. Gueniat, L. Mathelin, and L. Pastur. A dynamic mode decomposition approach for large and arbitrarily sampled systems. *Physics of Fluids*, 27(2):025113, 2015.
- [226] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on signal processing*, 50(2):425–437, 2002.
- [227] A. Haar. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, 69(3):331–371, 1910.
- [228] N. Halko, P.-G. Martinsson, Y. Shkolnisky, and M. Tygert. An algorithm for the principal component analysis of large data sets. *SIAM Journal on Scientific Computing*, 33:2580–2594, 2011.
- [229] N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.
- [230] N. Halko, P.-G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.
- [231] S. J. Hammarling. Numerical solution of the stable, non-negative definite Lyapunov equation. *IMA Journal of Numerical Analysis*, 2(3):303–323, 1982.
- [232] S. Han and B. Feeny. Application of proper orthogonal decomposition to structural vibration analysis. *Mechanical Systems and Signal Processing*, 17(5):989–1001, 2003.
- [233] N. Hansen, A. S. Niederberger, L. Guzzella, and P. Koumoutsakos. A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion. *IEEE Transactions on Evolutionary Computation*, 13(1):180–197, 2009.
- [234] D. Harrison Jr and D. L. Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management*, 5(1):81–102, 1978.
- [235] R. A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an “explanatory” multi-modal factor analysis. *UCLA working papers in phonetics*, 16:1–84, 1970. Available at <http://www.psychology.uwo.ca/faculty/harshman/wpppfacl0.pdf>.

- [236] T. Hastie, R. Tibshirani, J. Friedman, T. Hastie, J. Friedman, and R. Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.
- [237] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [238] M. Heath, A. Laub, C. Paige, and R. Ward. Computing the singular value decomposition of a product of two matrices. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1147–1159, 1986.
- [239] M. Heideman, D. Johnson, and C. Burrus. Gauss and the history of the fast Fourier transform. *IEEE ASSP Magazine*, 1(4):14–21, 1984.
- [240] W. Heisenberg. Über den anschaulichen inhalt der quantentheoretischen kinematik und mechanik. In *Original Scientific Papers Wissenschaftliche Originalarbeiten*, pages 478–504. Springer, 1985.
- [241] M. S. Hemati, C. W. Rowley, E. A. Deem, and L. N. Cattafesta. De-biasing the dynamic mode decomposition for applied Koopman spectral analysis. *Theoretical and Computational Fluid Dynamics*, 31(4):349–368, 2017.
- [242] M. S. Hemati, M. O. Williams, and C. W. Rowley. Dynamic mode decomposition for large and streaming datasets. *Physics of Fluids (1994-present)*, 26(11):111701, 2014.
- [243] K. K. Herrity, A. C. Gilbert, and J. A. Tropp. Sparse approximation via iterative thresholding. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 3, pages III–III. IEEE, 2006.
- [244] J. S. Hesthaven, G. Rozza, and B. Stamm. Certified reduced basis methods for parametrized partial differential equations. *SpringerBriefs in Mathematics*, 2015.
- [245] T. Hey, S. Tansley, K. M. Tolle, et al. *The fourth paradigm: data-intensive scientific discovery*, volume 1. Microsoft research Redmond, WA, 2009.
- [246] G. E. Hinton and T. J. Sejnowski. Learning and relearning in boltzmann machines. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1(282-317):2, 1986.
- [247] B. L. Ho and R. E. Kalman. Effective construction of linear state-variable models from input/output data. In *Proceedings of the 3rd Annual Allerton Conference on Circuit and System Theory*, pages 449–459, 1965.
- [248] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [249] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.
- [250] J. H. Holland. *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press, 1975.
- [251] P. Holmes and J. Guckenheimer. *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, volume 42 of *Applied Mathematical Sciences*. Springer-Verlag, Berlin, Heidelberg, 1983.
- [252] P. Holmes, J. L. Lumley, G. Berkooz, and C. W. Rowley. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press, Cambridge, 2nd paperback edition, 2012.

- [253] E. Hopf. The partial differential equation $u_t + uu_x = \mu u_{xx}$. *Communications on Pure and Applied mathematics*, 3(3):201–230, 1950.
- [254] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.
- [255] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.
- [256] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, Sept. 1933.
- [257] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:498–520, Oct. 1933.
- [258] C. Huang, W. E. Anderson, M. E. Harvazinski, and V. Sankaran. Analysis of self-excited combustion instabilities using decomposition techniques. In *51st AIAA Aerospace Sciences Meeting*, pages 1–18, 2013.
- [259] D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex. *Journal of Physiology*, 160:106–154, 1962.
- [260] P. J. Huber. Robust statistics. In *International Encyclopedia of Statistical Science*, pages 1248–1251. Springer, 2011.
- [261] S. J. Illingworth, A. S. Morgans, and C. W. Rowley. Feedback control of flow resonances using balanced reduced-order models. *Journal of Sound and Vibration*, 330(8):1567–1581, 2010.
- [262] E. Jacobsen and R. Lyons. The sliding DFT. *IEEE Signal Processing Magazine*, 20(2):74–80, 2003.
- [263] H. Jaeger and H. Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *science*, 304(5667):78–80, 2004.
- [264] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning*. Springer, 2013.
- [265] M. C. Johnson, S. L. Brunton, N. B. Kundtz, and J. N. Kutz. Extremum-seeking control of a beam pattern of a reconfigurable holographic metamaterial antenna. *Journal of the Optical Society of America A*, 33(1):59–68, 2016.
- [266] R. A. Johnson and D. Wichern. *Multivariate analysis*. Wiley Online Library, 2002.
- [267] W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics*, 26(189–206):1, 1984.
- [268] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.
- [269] S. Joshi and S. Boyd. Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, 57(2):451–462, 2009.
- [270] M. R. Jovanović, P. J. Schmid, and J. W. Nichols. Sparsity-promoting dynamic mode decomposition. *Physics of Fluids*, 26(2):024103, 2014.
- [271] J. N. Juang. *Applied System Identification*. Prentice Hall PTR, Upper Saddle River, New Jersey, 1994.
- [272] J. N. Juang and R. S. Pappa. An eigensystem realization algorithm for modal parameter identification and model reduction. *Journal of Guidance, Control, and Dynamics*, 8(5):620–627, 1985.

- [273] J. N. Juang, M. Phan, L. G. Horta, and R. W. Longman. Identification of observer/Kalman filter Markov parameters: Theory and experiments. Technical Memorandum 104069, NASA, 1991.
- [274] S. J. Julier and J. K. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In *Int. symp. aerospace/defense sensing, simul. and controls*, volume 3, pages 182–193. Orlando, FL, 1997.
- [275] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.
- [276] E. Kaiser, J. N. Kutz, and S. L. Brunton. Data-driven discovery of Koopman eigenfunctions for control. *arXiv preprint arXiv:1707.01146*, 2017.
- [277] E. Kaiser, J. N. Kutz, and S. L. Brunton. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society of London A*, 474(2219), 2018.
- [278] E. Kaiser, B. R. Noack, L. Cordier, A. Spohn, M. Segond, M. Abel, G. Daviller, J. Östh, S. Krajnović, and R. K. Niven. Cluster-based reduced-order modelling of a mixing layer. *Journal of Fluid Mechanics*, 754:365–414, 2014.
- [279] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45, 1960.
- [280] K. Karhunen. Über lineare methoden in der wahrscheinlichkeitsrechnung, vol. 37. *Annales AcademiæScientiarum Fennicæ, Ser. A. I*, 1947.
- [281] K. Kasper, L. Mathelin, and H. Abou-Kandil. A machine learning approach for constrained sensor placement. In *American Control Conference (ACC), 2015*, pages 4479–4484. IEEE, 2015.
- [282] A. K. Kassam and L. N. Trefethen. Fourth-order time-stepping for stiff PDEs. *SIAM Journal on Scientific Computing*, 26(4):1214–1233, 2005.
- [283] M. Kearns and L. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM (JACM)*, 41(1):67–95, 1994.
- [284] A. R. Kellem, S. Chaturantabut, D. C. Sorensen, and S. J. Cox. Morphologically accurate reduced order modeling of spiking neurons. *Journal of computational neuroscience*, 28(3):477–494, 2010.
- [285] J. Kepler. *Tabulae Rudolphinae, quibus Astronomicae scientiae, temporum longinquitate collapsae Restauratio continetur*. Ulm: Jonas Saur, 1627.
- [286] G. Kerschen and J.-C. Golinval. Physical interpretation of the proper orthogonal modes using the singular value decomposition. *Journal of Sound and Vibration*, 249(5):849–865, 2002.
- [287] G. Kerschen, J.-c. Golinval, A. F. Vakakis, and L. A. Bergman. The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview. *Nonlinear dynamics*, 41(1-3):147–169, 2005.
- [288] I. G. Kevrekidis, C. W. Gear, J. M. Hyman, P. G. Kevrekidis, O. Runborg, and C. Theodoropoulos. Equation-free, coarse-grained multiscale computation: Enabling microscopic simulators to perform system-level analysis. *Communications in Mathematical Science*, 1(4):715–762, 2003.
- [289] N. J. Killingsworth and M. Krstic. PID tuning using extremum seeking: online, model-free performance optimization. *IEEE Control Systems Magazine*, February:70–79, 2006.

- [290] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [291] M. Kirby and L. Sirovich. Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 12(1):103–108, 1990.
- [292] V. C. Klema and A. J. Laub. The singular value decomposition: Its computation and some applications. *IEEE Transactions on Automatic Control*, 25(2):164–176, 1980.
- [293] S. Klus, F. Nüske, P. Koltai, H. Wu, I. Kevrekidis, C. Schütte, and F. Noé. Data-driven model reduction and transfer operator approximation. *Journal of Nonlinear Science*, pages 1–26, 2018.
- [294] R. Koch. *The 80/20 Principle*. Nicholas Brealey Publishing, 1997.
- [295] R. Koch. *Living the 80/20 way*. Audio-Tech Business Book Summaries, Incorporated, 2006.
- [296] R. Koch. *The 80/20 principle: the secret to achieving more with less*. Crown Business, 2011.
- [297] R. Koch. *The 80/20 principle and 92 other powerful laws of nature: the science of success*. Nicholas Brealey Publishing, 2013.
- [298] T. Kohonen. The self-organizing map. *Neurocomputing*, 21(1-3):1–6, 1998.
- [299] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, September 2009.
- [300] B. O. Koopman. Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.
- [301] B. O. Koopman and J.-v. Neumann. Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences of the United States of America*, 18(3):255, 1932.
- [302] M. Korda and I. Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93(149–160), 2018.
- [303] M. Korda and I. Mezić. On convergence of extended dynamic mode decomposition to the Koopman operator. *Journal of Nonlinear Science*, 28(2):687–710, 2018.
- [304] P. Koumoutsakos, J. Freund, and D. Parekh. Evolution strategies for automatic optimization of jet mixing. *AIAA journal*, 39(5):967–969, 2001.
- [305] K. Kowalski, W.-H. Steeb, and K. Kowalksi. *Nonlinear dynamical systems and Carleman linearization*. World Scientific, 1991.
- [306] J. R. Koza. *Genetic programming: on the programming of computers by means of natural selection*, volume 1. MIT press, 1992.
- [307] J. R. Koza, F. H. Bennett III, and O. Stiffelman. Genetic programming as a darwinian invention machine. In *Genetic Programming*, pages 93–108. Springer, 1999.
- [308] B. Kramer, P. Grover, P. Boufounos, M. Benosman, and S. Nabi. Sparse sensing and dmd based identification of flow regimes and bifurcations in complex flows. *SIAM J. Appl. Dyn. Syst.*, 16(2):1164–1196, 2017.
- [309] J. P. Krieger and M. Krstic. Extremum seeking based on atmospheric turbulence for aircraft endurance. *Journal of Guidance, Control, and Dynamics*, 34(6):1876–1885, 2011.
- [310] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

- [311] M. Krstic, A. Krupadanam, and C. Jacobson. Self-tuning control of a nonlinear model of combustion instabilities. *IEEE Tr. Contr. Syst. Technol.*, 7(4):424–436, 1999.
- [312] M. Krstić and H. Wang. Stability of extremum seeking feedback for general nonlinear dynamic systems. *Automatica*, 36:595–601, 2000.
- [313] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *Advances in Neural Information Processing Systems*, pages 2539–2547, 2015.
- [314] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.
- [315] K. Kunisch and S. Volkwein. Optimal snapshot location for computing pod basis functions. *ESAIM: Mathematical Modelling and Numerical Analysis*, 44(3):509–529, 2010.
- [316] J. N. Kutz. *Data-Driven Modeling & Scientific Computation: Methods for Complex Systems & Big Data*. Oxford University Press, 2013.
- [317] J. N. Kutz, S. L. Brunton, B. W. Brunton, and J. L. Proctor. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*. SIAM, 2016.
- [318] J. N. Kutz, X. Fu, and S. L. Brunton. Multi-resolution dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 15(2):713–735, 2016.
- [319] J. N. Kutz, S. Sargsyan, and S. L. Brunton. Leveraging sparsity and compressive sensing for reduced order modeling. In *Model Reduction of Parametrized Systems*, pages 301–315. Springer, 2017.
- [320] S. Lall, J. E. Marsden, and S. Glavaški. Empirical model reduction of controlled nonlinear systems. In *IFAC World Congress*, volume F, pages 473–478. International Federation of Automatic Control, 1999.
- [321] S. Lall, J. E. Marsden, and S. Glavaški. A subspace approach to balanced truncation for model reduction of nonlinear control systems. *International Journal of Robust and Nonlinear Control*, 12(6):519–535, 2002.
- [322] Y. Lan and I. Mezić. Linearization in the large of nonlinear systems and Koopman operator spectrum. *Physica D: Nonlinear Phenomena*, 242(1):42–53, 2013.
- [323] A. Laub. A Schur method for solving algebraic Riccati equations. *IEEE Transactions on automatic control*, 24(6):913–921, 1979.
- [324] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.
- [325] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [326] J. H. Lee. Model predictive control: Review of the three decades of development. *International Journal of Control, Automation and Systems*, 9(3):415–424, 2011.
- [327] K. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(5):684–698, 2005.
- [328] A. M. Legendre. *Nouvelles méthodes pour la détermination des orbites des comètes*. F. Didot, 1805.
- [329] V. Lenaerts, G. Kerschen, and J.-C. Golinval. Proper orthogonal decomposition for model updating of non-linear mechanical systems. *Mechanical Systems and Signal Processing*, 15(1):31–43, 2001.

- [330] I. Lenz, R. A. Knepper, and A. Saxena. Deepmpc: Learning deep latent features for model predictive control. In *Robotics: Science and Systems*, 2015.
- [331] R. Leyva, C. Alonso, I. Queinnec, A. Cid-Pastor, D. Lagrange, and L. Martinez-Salamero. MPPT of photovoltaic systems using extremum-seeking control. *Ieee Transactions On Aerospace and Electronic Systems*, 42(1):249–258, Jan. 2006.
- [332] Q. Li, F. Dietrich, E. M. Boltt, and I. G. Kevrekidis. Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 27(10):103111, 2017.
- [333] Y. Liang, H. Lee, S. Lim, W. Lin, K. Lee, and C. Wu. Proper orthogonal decomposition and its applications- part i: Theory. *Journal of Sound and vibration*, 252(3):527–544, 2002.
- [334] E. Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 581–588. ACM, 2013.
- [335] E. Liberty, F. Woolfe, P.-G. Martinsson, V. Rokhlin, and M. Tygert. Randomized algorithms for the low-rank approximation of matrices. *Proceedings of the National Academy of Sciences*, 104:20167–20172, 2007.
- [336] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.
- [337] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.
- [338] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, 1999.
- [339] S. Lloyd. Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2):129–137, 1982.
- [340] M. Loeve. *Probability Theory*. Van Nostrand, Princeton, NJ, 1955.
- [341] J.-C. Loiseau and S. L. Brunton. Constrained sparse Galerkin regression. *Journal of Fluid Mechanics*, 838:42–67, 2018.
- [342] J.-C. Loiseau, B. R. Noack, and S. L. Brunton. Sparse reduced-order modeling: sensor-based dynamics to full-state estimation. *Journal of Fluid Mechanics*, 844:459–490, 2018.
- [343] R. W. Longman. Iterative learning control and repetitive control for engineering practice. *International journal of control*, 73(10):930–954, 2000.
- [344] E. N. Lorenz. Empirical orthogonal functions and statistical weather prediction. Technical report, Massachusetts Institute of Technology, Dec. 1956.
- [345] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, 20(2):130–141, 1963.
- [346] D. M. Luchtenburg and C. W. Rowley. Model reduction using snapshot-based realizations. *Bulletin of the American Physical Society*, 56, 2011.
- [347] J. Lumley. Toward a turbulent constitutive relation. *Journal of Fluid Mechanics*, 41(02):413–434, 1970.
- [348] B. Lusch, E. C. Chi, and J. N. Kutz. Shape constrained tensor decompositions using sparse representations in over-complete libraries. *arXiv preprint arXiv:1608.04674*, 2016.

- [349] B. Lusch, J. N. Kutz, and S. L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, 2018.
- [350] F. Lusseyran, F. Gueniat, J. Basley, C. L. Douay, L. R. Pastur, T. M. Faure, and P. J. Schmid. Flow coherent structures and frequency signature: application of the dynamic modes decomposition to open cavity flow. In *Journal of Physics: Conference Series*, volume 318, page 042036. IOP Publishing, 2011.
- [351] Z. Ma, S. Ahuja, and C. W. Rowley. Reduced order models for control of fluids using the eigensystem realization algorithm. *Theor. Comput. Fluid Dyn.*, 25(1):233–247, 2011.
- [352] W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560, 2002.
- [353] A. Mackey, H. Schaeffer, and S. Osher. On the compressive spectral method. *Multiscale Modeling & Simulation*, 12(4):1800–1827, 2014.
- [354] M. W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3:123–224, 2011.
- [355] A. J. Majda and J. Harlim. Physics constrained nonlinear regression models for time series. *Nonlinearity*, 26(1):201, 2012.
- [356] A. J. Majda and Y. Lee. Conceptual dynamical models for turbulence. *Proceedings of the National Academy of Sciences*, 111(18):6548–6553, 2014.
- [357] S. Mallat. *A wavelet tour of signal processing*. Academic press, 1999.
- [358] S. Mallat. Understanding deep convolutional networks. *Phil. Trans. R. Soc. A*, 374(2065):20150203, 2016.
- [359] S. G. Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7):674–693, 1989.
- [360] J. Mandel. Use of the singular value decomposition in regression analysis. *The American Statistician*, 36(1):15–24, 1982.
- [361] N. M. Mangan, S. L. Brunton, J. L. Proctor, and J. N. Kutz. Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Transactions on Molecular, Biological, and Multi-Scale Communications*, 2(1):52–63, 2016.
- [362] N. M. Mangan, J. N. Kutz, S. L. Brunton, and J. L. Proctor. Model selection for dynamical systems via sparse regression and information criteria. *Proceedings of the Royal Society A*, 473(2204):1–16, 2017.
- [363] J. Mann and J. N. Kutz. Dynamic mode decomposition for financial trading strategies. *Quantitative Finance*, pages 1–13, 2016.
- [364] K. Manohar, B. W. Brunton, J. N. Kutz, and S. L. Brunton. Data-driven sparse sensor placement. *Invited for IEEE Control Systems Magazine*, 2017.
- [365] K. Manohar, S. L. Brunton, and J. N. Kutz. Environmental identification in flight using sparse approximation of wing strain. *Journal of Fluids and Structures*, 70:162–180, 2017.
- [366] K. Manohar, E. Kaiser, S. L. Brunton, and J. N. Kutz. Optimized sampling for multiscale dynamics. *SIAM Multiscale modeling and simulation*, 17(1):117–136, 2019.
- [367] K. Manohar, J. N. Kutz, and S. L. Brunton. Optimized sensor and actuator placement for balanced models. *arXiv preprint arXiv:1812.01574*, 2018.

- [368] A. Mardt, L. Pasquali, H. Wu, and F. Noé. VAMPnets: Deep learning of molecular kinetics. *Nature Communications*, 9(5), 2018.
- [369] J. E. Marsden and T. S. Ratiu. *Introduction to mechanics and symmetry*. Springer-Verlag, 2nd edition, 1999.
- [370] P.-G. Martinsson. Randomized methods for matrix computations and analysis of high dimensional data. *arXiv preprint arXiv:1607.01649*, 2016.
- [371] P.-G. Martinsson, V. Rokhlin, and M. Tygert. A randomized algorithm for the decomposition of matrices. *Applied and Computational Harmonic Analysis*, 30:47–68, 2011.
- [372] J. L. Maryak, J. C. Spall, and B. D. Heydon. Use of the Kalman filter for inference in state-space models with unknown noise distributions. *IEEE Transactions on Automatic Control*, 49(1):87–90, 2004.
- [373] L. Massa, R. Kumar, and P. Ravindran. Dynamic mode decomposition analysis of detonation waves. *Physics of Fluids (1994-present)*, 24(6):066101, 2012.
- [374] L. Mathelin, K. Kasper, and H. Abou-Kandil. Observable dictionary learning for high-dimensional statistical inference. *Archives of Computational Methods in Engineering*, 25(1):103–120, 2018.
- [375] R. Maury, M. Keonig, L. Cattafesta, P. Jordan, and J. Delville. Extremum-seeking control of jet noise. *Aeroacoustics*, 11(3&4):459–474, 2012.
- [376] I. Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1-3):309–325, 2005.
- [377] I. Mezić. Analysis of fluid flows via spectral properties of the Koopman operator. *Ann. Rev. Fluid Mech.*, 45:357–378, 2013.
- [378] I. Mezić. *Spectral operator methods in dynamical systems: Theory and applications*. Springer, 2017.
- [379] I. Mezić and A. Banaszuk. Comparison of systems with complex behavior. *Physica D: Nonlinear Phenomena*, 197(1):101–133, 2004.
- [380] I. Mezić and S. Wiggins. A method for visualization of invariant sets of dynamical systems based on the ergodic partition. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 9(1):213–218, 1999.
- [381] M. Milano and P. Koumoutsakos. Neural network modeling for near wall turbulent flow. *Journal of Computational Physics*, 182(1):1–26, 2002.
- [382] T. M. Mitchell. *Machine Learning*. McGraw Hill, 1997.
- [383] Y. Mizuno, D. Duke, C. Atkinson, and J. Soria. Investigation of wall-bounded turbulent flow using dynamic mode decomposition. In *Journal of Physics: Conference Series*, volume 318, page 042040. IOP Publishing, 2011.
- [384] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1928–1937, 2016.
- [385] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

- [386] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.
- [387] J. P. Moeck, J.-F. Bourgouin, D. Durox, T. Schuller, and S. Candel. Tomographic reconstruction of heat release rate perturbations induced by helical modes in turbulent swirl flames. *Experiments in Fluids*, 54(4):1–17, 2013.
- [388] B. C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, AC-26(1):17–32, 1981.
- [389] C. C. Moore. Ergodic theorem, ergodic theory, and statistical mechanics. *Proceedings of the National Academy of Sciences*, 112(7):1907–1911, 2015.
- [390] K. L. Moore. *Iterative learning control for deterministic systems*. Springer Science & Business Media, 2012.
- [391] M. Morari and J. H. Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4):667–682, 1999.
- [392] T. W. Muld, G. Efraimsson, and D. S. Henningson. Flow structures around a high-speed train extracted using proper orthogonal decomposition and dynamic mode decomposition. *Computers & Fluids*, 57:87–97, 2012.
- [393] T. W. Muld, G. Efraimsson, and D. S. Henningson. Mode decomposition on surface-mounted cube. *Flow, Turbulence and Combustion*, 88(3):279–310, 2012.
- [394] S. Müller, M. Milano, and P. Koumoutsakos. Application of machine learning algorithms to flow modeling and optimization. *Annual Research Briefs*, pages 169–178, 1999.
- [395] I. Munteanu, A. I. Bratu, and E. Ceanga. Wind turbulence used as searching signal for MPPT in variable-speed wind energy conversion systems. *Renewable Energy*, 34(1):322–327, Jan. 2009.
- [396] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [397] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.
- [398] D. Needell and J. A. Tropp. CoSaMP: iterative signal recovery from incomplete and inaccurate samples. *Communications of the ACM*, 53(12):93–100, 2010.
- [399] J. v. Neumann. Proof of the quasi-ergodic hypothesis. *Proceedings of the National Academy of Sciences*, 18(1):70–82, 1932.
- [400] N. Nguyen, A. Patera, and J. Peraire. A best points interpolation method for efficient approximation of parametrized functions. *International Journal for Numerical Methods in Engineering*, 73(4):521–543, 2008.
- [401] Y. Nievergelt and Y. Nievergelt. *Wavelets made easy*, volume 174. Springer, 1999.
- [402] B. R. Noack, K. Afanasiev, M. Morzynski, G. Tadmor, and F. Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335–363, 2003.
- [403] B. R. Noack, T. Duriez, L. Cordier, M. Segond, M. Abel, S. L. Brunton, M. Morzyński, J.-C. Laurentie, V. Parezanovic, and J.-P. Bonnet. Closed-loop turbulence control with machine learning methods. *Bulletin Am. Phys. Soc.*, 58(18):M25.0009, p. 418, 2013.

- [404] B. R. Noack, M. Morzynski, and G. Tadmor. *Reduced-order modelling for flow control*, volume 528. Springer Science & Business Media, 2011.
- [405] F. Noé and F. Nüske. A variational approach to modeling slow processes in stochastic dynamical systems. *Multiscale Modeling & Simulation*, 11(2):635–655, 2013.
- [406] E. Noether. Invariante variationsprobleme nachr. d. könig. gesellsch. d. wiss. zu göttingen, math-phys. klasse 1918: 235-257. English Reprint: physics/0503066, <http://dx.doi.org/10.1080/00411457108231446>, page 57, 1918.
- [407] F. Nüske, B. G. Keller, G. Pérez-Hernández, A. S. Mey, and F. Noé. Variational approach to molecular kinetics. *Journal of chemical theory and computation*, 10(4):1739–1752, 2014.
- [408] F. Nüske, R. Schneider, F. Vitalini, and F. Noé. Variational tensor approach for approximating the rare-event kinetics of macromolecular systems. *J. Chem. Phys.*, 144(5):054105, 2016.
- [409] H. Nyquist. Certain topics in telegraph transmission theory. *Transactions of the A. I. E. E.*, pages 617–644, FEB 1928.
- [410] G. Obinata and B. D. Anderson. *Model reduction for control system design*. Springer Science & Business Media, 2012.
- [411] C. M. Ostoich, D. J. Bodony, and P. H. Geubelle. Interaction of a Mach 2.25 turbulent boundary layer with a fluttering panel using direct numerical simulation. *Physics of Fluids (1994-present)*, 25(11):110806, 2013.
- [412] S. E. Otto and C. W. Rowley. Linearly-recurrent autoencoder networks for learning dynamics. *arXiv preprint arXiv:1712.01378*, 2017.
- [413] Y. Ou, C. Xu, E. Schuster, T. C. Luce, J. R. Ferron, M. L. Walker, and D. A. Humphreys. Design and simulation of extremum-seeking open-loop optimal control of current profile in the DIII-D tokamak. *Plasma Physics and Controlled Fusion*, 50:115001–1–115001–24, 2008.
- [414] V. Ozoliņš, R. Lai, R. Caflisch, and S. Osher. Compressed modes for variational problems in mathematics and physics. *Proceedings of the National Academy of Sciences*, 110(46):18368–18373, 2013.
- [415] C. Pan, D. Yu, and J. Wang. Dynamical mode decomposition of Gurney flap wake flow. *Theoretical and Applied Mechanics Letters*, 1(1):012002, 2011.
- [416] V. Parezanović, T. Duriez, L. Cordier, B. R. Noack, J. Delville, J.-P. Bonnet, M. Segond, M. Abel, and S. L. Brunton. Closed-loop control of an experimental mixing layer using machine learning control. *arXiv preprint arXiv:1408.3259*, 2014.
- [417] V. Parezanovic, J.-C. Laurentie, T. Duriez, C. Fourment, J. Delville, J.-P. Bonnet, L. Cordier, B. R. Noack, M. Segond, M. Abel, T. Shaqarin, and S. L. Brunton. Mixing layer manipulation experiment – from periodic forcing to machine learning closed-loop control. *Journal Flow Turbulence and Combustion*, 94(1):155–173, 2015.
- [418] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2(7–12):559–572, 1901.
- [419] B. Peherstorfer, D. Butnaru, K. Willcox, and H.-J. Bungartz. Localized discrete empirical interpolation method. *SIAM Journal on Scientific Computing*, 36(1):A168–A192, 2014.
- [420] B. Peherstorfer and K. Willcox. Detecting and adapting to parameter changes for reduced models of dynamic data-driven application systems. *Procedia Computer Science*, 51:2553–2562, 2015.

- [421] B. Peherstorfer and K. Willcox. Dynamic data-driven reduced-order models. *Computer Methods in Applied Mechanics and Engineering*, 291:21–41, 2015.
- [422] B. Peherstorfer and K. Willcox. Online adaptive model reduction for nonlinear systems via low-rank updates. *SIAM Journal on Scientific Computing*, 37(4):A2123–A2150, 2015.
- [423] S. Peitz and S. Klus. Koopman operator-based model reduction for switched-system control of PDEs. *arXiv preprint arXiv:1710.06759*, 2017.
- [424] S. D. Pendergrass, J. N. Kutz, and S. L. Brunton. Streaming GPU singular value and dynamic mode decompositions. *arXiv preprint arXiv:1612.07875*, 2016.
- [425] R. Penrose. A generalized inverse for matrices. In *Mathematical proceedings of the Cambridge philosophical society*, volume 51, pages 406–413. Cambridge Univ Press, 1955.
- [426] R. Penrose and J. A. Todd. On best approximate solutions of linear matrix equations. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 52, pages 17–19. Cambridge Univ Press, 1956.
- [427] L. Perko. *Differential equations and dynamical systems*, volume 7. Springer Science & Business Media, 2013.
- [428] M. Phan, L. G. Horta, J. N. Juang, and R. W. Longman. Linear system identification via an asymptotically stable observer. *Journal of Optimization Theory and Applications*, 79:59–86, 1993.
- [429] M. A. Pinsky. *Introduction to Fourier analysis and wavelets*, volume 102. American Mathematical Soc., 2002.
- [430] T. Poggio. Deep learning: mathematics and neuroscience. *Views & Reviews, McGovern Center for Brains, Minds and Machines*, pages 1–7, 2016.
- [431] P. Poncet, G.-H. Cottet, and P. Koumoutsakos. Control of three-dimensional wakes using evolution strategies. *Comptes Rendus Mecanique*, 333(1):65–77, 2005.
- [432] C. Poultney, S. Chopra, Y. L. Cun, et al. Efficient learning of sparse representations with an energy-based model. In *Advances in neural information processing systems*, pages 1137–1144, 2007.
- [433] J. L. Proctor, S. L. Brunton, B. W. Brunton, and J. N. Kutz. Exploiting sparsity and equation-free architectures in complex systems (invited review). *The European Physical Journal Special Topics*, 223(13):2665–2684, 2014.
- [434] J. L. Proctor, S. L. Brunton, and J. N. Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016.
- [435] J. L. Proctor and P. A. Eckhoff. Discovering dynamic patterns from infectious disease data using dynamic mode decomposition. *International health*, 7(2):139–145, 2015.
- [436] H. Qi and S. M. Hughes. Invariance of principal components under low-dimensional random projection of the data. IEEE International Conference on Image Processing, October 2012.
- [437] S. Qian and D. Chen. Discrete Gabor transform. *IEEE transactions on signal processing*, 41(7):2429–2438, 1993.
- [438] S. J. Qin and T. A. Badgwell. An overview of industrial model predictive control technology. In *AIChe Symposium Series*, volume 93, pages 232–256, 1997.
- [439] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003.

- [440] Q. Qu, J. Sun, and J. Wright. Finding a sparse vector in a subspace: Linear sparsity using alternating directions. In *Advances in Neural Information Processing Systems 27*, pages 3401—3409, 2014.
- [441] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*, volume 92. Springer, 2015.
- [442] A. Quarteroni and G. Rozza. *Reduced Order Methods for Modeling and Computational Reduction*, volume 9 of *MS&A – Modeling, Simulation & Applications*. Springer, 2013.
- [443] J. R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.
- [444] J. R. Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.
- [445] M. Raissi and G. E. Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 357:125–141, 2018.
- [446] C. R. Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):159–203, 1948.
- [447] J. B. Rawlings. Tutorial overview of model predictive control. *IEEE Control Systems*, 20(3):38–52, 2000.
- [448] S. Raychaudhuri, J. M. Stuart, and R. B. Altman. Principal components analysis to summarize microarray experiments: application to sporulation time series. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, page 455. NIH Public Access, 2000.
- [449] R. H. Reichle, D. B. McLaughlin, and D. Entekhabi. Hydrologic data assimilation with the ensemble Kalman filter. *Monthly Weather Review*, 130(1):103–114, 2002.
- [450] B. Ren, P. Frihauf, R. J. Rafac, and M. Krstić. Laser pulse shaping via extremum seeking. *Control Engineering Practice*, 20:674–683, 2012.
- [451] B. Ristic, S. Arulampalam, and N. J. Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Artech house, 2004.
- [452] A. J. Roberts. *Model emergent dynamics in complex systems*. SIAM, 2014.
- [453] C. A. Rohde. Generalized inverses of partitioned matrices. *Journal of the Society for Industrial & Applied Mathematics*, 13(4):1033–1035, 1965.
- [454] V. Rokhlin, A. Szlam, and M. Tygert. A randomized algorithm for principal component analysis. *SIAM Journal on Matrix Analysis and Applications*, 31:1100–1124, 2009.
- [455] C. Rowley. Model reduction for fluids using balanced proper orthogonal decomposition. *Int. J. Bifurcation and Chaos*, 15(3):997–1013, 2005.
- [456] C. W. Rowley, T. Colonius, and R. M. Murray. Model reduction for compressible flows using POD and Galerkin projection. *Physica D*, 189:115–129, 2004.
- [457] C. W. Rowley and J. E. Marsden. Reconstruction equations and the Karhunen–Loève expansion for systems with symmetry. *Physica D: Nonlinear Phenomena*, 142(1):1–19, 2000.
- [458] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. Henningson. Spectral analysis of nonlinear flows. *J. Fluid Mech.*, 645:115–127, 2009.
- [459] S. Roy, J.-C. Hua, W. Barnhill, G. H. Gunaratne, and J. R. Gord. Deconvolution of reacting-flow dynamics using proper orthogonal and dynamic mode decompositions. *Physical Review E*, 91(1):013001, 2015.

- [460] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(e1602614), 2017.
- [461] T. P. Sapsis and A. J. Majda. Statistically accurate low-order models for uncertainty quantification in turbulent dynamical systems. *Proceedings of the National Academy of Sciences*, 110(34):13705–13710, 2013.
- [462] S. Sargsyan, S. L. Brunton, and J. N. Kutz. Nonlinear model reduction for dynamical systems using sparse sensor locations from learned libraries. *Physical Review E*, 92(033304), 2015.
- [463] S. Sarkar, S. Ganguly, A. Dalal, P. Saha, and S. Chakraborty. Mixed convective flow stability of nanofluids past a square cylinder by dynamic mode decomposition. *International Journal of Heat and Fluid Flow*, 44:624–634, 2013.
- [464] T. Sarlos. Improved approximation algorithms for large matrices via random projections. In *Foundations of Computer Science. 47th Annual IEEE Symposium on*, pages 143–152, 2006.
- [465] T. Sayadi and P. J. Schmid. Parallel data-driven decomposition algorithm for large-scale datasets: with application to transitional boundary layers. *Theoretical and Computational Fluid Dynamics*, pages 1–14, 2016.
- [466] T. Sayadi, P. J. Schmid, J. W. Nichols, and P. Moin. Reduced-order representation of near-wall structures in the late transitional boundary layer. *Journal of Fluid Mechanics*, 748:278–301, 2014.
- [467] H. Schaeffer. Learning partial differential equations via data discovery and sparse optimization. In *Proc. R. Soc. A*, volume 473, page 20160446. The Royal Society, 2017.
- [468] H. Schaeffer, R. Caflisch, C. D. Hauck, and S. Osher. Sparse dynamics for partial differential equations. *Proceedings of the National Academy of Sciences USA*, 110(17):6634–6639, 2013.
- [469] H. Schaeffer and S. G. McCalla. Sparse model selection via integral terms. *Physical Review E*, 96(2):023302, 2017.
- [470] R. E. Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.
- [471] M. Schlegel, B. R. Noack, and G. Tadmor. Low-dimensional Galerkin models and control of transitional channel flow. Technical Report 01/2004, Hermann-Föttinger-Institut für Strömungsmechanik, Technische Universität Berlin, Germany, 2004.
- [472] P. J. Schmid. Dynamic mode decomposition for numerical and experimental data. *J. Fluid. Mech.*, 656:5–28, 2010.
- [473] P. J. Schmid, L. Li, M. P. Juniper, and O. Pust. Applications of the dynamic mode decomposition. *Theoretical and Computational Fluid Dynamics*, 25(1-4):249–259, 2011.
- [474] P. J. Schmid and J. Sesterhenn. Dynamic mode decomposition of numerical and experimental data. In *61st Annual Meeting of the APS Division of Fluid Dynamics*. American Physical Society, Nov. 2008.
- [475] P. J. Schmid, D. Violato, and F. Scarano. Decomposition of time-resolved tomographic PIV. *Experiments in Fluids*, 52:1567–1579, 2012.
- [476] E. Schmidt. Zur theorie der linearen und nichtlinearen integralgleichungen. i teil. entwicklung willkürlichen funktionen nach system vorgescriebener. *Math. Ann.*, 3:433–476, 1907.

- [477] M. Schmidt and H. Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.
- [478] M. D. Schmidt, R. R. Vallabhanayula, J. W. Jenkins, J. E. Hood, A. S. Soni, J. P. Wikswo, and H. Lipson. Automated refinement and inference of analytical models for metabolic networks. *Physical biology*, 8(5):055011, 2011.
- [479] B. Schölkopf and A. J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.
- [480] G. Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.
- [481] A. Seena and H. J. Sung. Dynamic mode decomposition of turbulent cavity flows for self-sustained oscillations. *International Journal of Heat and Fluid Flow*, 32(6):1098–1110, 2011.
- [482] E. Sejdić, I. Djurović, and J. Jiang. Time–frequency feature representation using energy concentration: An overview of recent advances. *Digital Signal Processing*, 19(1):153–183, 2009.
- [483] O. Semeraro, G. Bellani, and F. Lundell. Analysis of time-resolved PIV measurements of a confined turbulent jet using POD and Koopman modes. *Experiments in Fluids*, 53(5):1203–1220, 2012.
- [484] O. Semeraro, F. Lusseyran, L. Pastur, and P. Jordan. Qualitative dynamics of wavepackets in turbulent jets. *Physical Review Fluids*, 2(094605), 2017.
- [485] G. Shabat, Y. Shmueli, Y. Aizenbud, and A. Averbuch. Randomized LU decomposition. *Applied and Computational Harmonic Analysis*, 2016.
- [486] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.
- [487] A. S. Sharma, I. Mezić, and B. J. McKeon. Correspondence between Koopman mode decomposition, resolvent mode decomposition, and invariant solutions of the Navier-Stokes equations. *Physical Review Fluids*, 1(3):032402, 2016.
- [488] E. Shlizerman, E. Ding, M. O. Williams, and J. N. Kutz. The proper orthogonal decomposition for dimensionality reduction in mode-locked lasers and optical systems. *International Journal of Optics*, 2012, 2011.
- [489] V. Simoncini. A new iterative method for solving large-scale Lyapunov matrix equations. *SIAM Journal on Scientific Computing*, 29(3):1268–1288, 2007.
- [490] L. Sirovich. Turbulence and the dynamics of coherent structures, parts I-III. *Q. Appl. Math.*, XLV(3):561–590, 1987.
- [491] L. Sirovich and M. Kirby. A low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4(3):519–524, 1987.
- [492] S. Skogestad and I. Postlethwaite. *Multivariable feedback control*. Wiley, Chichester, 1996.
- [493] P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. Technical report, COLORADO UNIV AT BOULDER DEPT OF COMPUTER SCIENCE, 1986.
- [494] G. Solari, L. Carassale, and F. Tubino. Proper orthogonal decomposition in wind engineering. part 1: A state-of-the-art and some prospects. *Wind and Structures*, 10(2):153–176, 2007.

- [495] G. Song, F. Alizard, J.-C. Robinet, and X. Gloerfelt. Global and Koopman modes analysis of sound generation in mixing layers. *Physics of Fluids (1994-present)*, 25(12):124101, 2013.
- [496] D. C. Sorensen and Y. Zhou. Direct methods for matrix Sylvester and Lyapunov equations. *Journal of Applied Mathematics*, 2003(6):277–303, 2003.
- [497] M. Sorokina, S. Sygletos, and S. Turitsyn. Sparse identification for nonlinear optical communication systems: SINO method. *Optics express*, 24(26):30433–30443, 2016.
- [498] J. C. Spall. The Kantorovich inequality for error analysis of the Kalman filter with unknown noise distributions. *Automatica*, 31(10):1513–1517, 1995.
- [499] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [500] W.-H. Steeb and F. Wilhelm. Non-linear autonomous systems of differential equations and Carleman linearization procedure. *Journal of Mathematical Analysis and Applications*, 77(2):601–611, 1980.
- [501] R. F. Stengel. *Optimal control and estimation*. Courier Corporation, 2012.
- [502] G. W. Stewart. On the early history of the singular value decomposition. *SIAM review*, 35(4):551–566, 1993.
- [503] G. Sugihara, R. May, H. Ye, C.-h. Hsieh, E. Deyle, M. Fogarty, and S. Munch. Detecting causality in complex ecosystems. *Science*, 338(6106):496–500, 2012.
- [504] A. Surana. Koopman operator based observer synthesis for control-affine nonlinear systems. In *55th IEEE Conference on Decision and Control (CDC)*, pages 6492–6499, 2016.
- [505] A. Surana and A. Banaszuk. Linear observer synthesis for nonlinear systems using Koopman operator framework. *IFAC-PapersOnLine*, 49(18):716–723, 2016.
- [506] Y. Susuki and I. Mezić. A prony approximation of Koopman mode decomposition. In *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*, pages 7022–7027. IEEE, 2015.
- [507] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.
- [508] A. Svenkeson, B. Glaz, S. Stanton, and B. J. West. Spectral decomposition of nonlinear systems with memory. *Phys. Rev. E*, 93:022211, Feb 2016.
- [509] S. Svoronos, D. Papageorgiou, and C. Tsiligiannis. Discretization of nonlinear control systems via the Carleman linearization. *Chemical engineering science*, 49(19):3263–3267, 1994.
- [510] D. L. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18(8):831–836, 1996.
- [511] K. Taira and T. Colonius. The immersed boundary method: a projection approach. *Journal of Computational Physics*, 225(2):2118–2137, 2007.
- [512] N. Takeishi, Y. Kawahara, Y. Tabei, and T. Yairi. Bayesian dynamic mode decomposition. *Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017.
- [513] N. Takeishi, Y. Kawahara, and T. Yairi. Learning Koopman invariant subspaces for dynamic mode decomposition. In *Advances in Neural Information Processing Systems*, pages 1130–1140, 2017.

- [514] N. Takeishi, Y. Kawahara, and T. Yairi. Subspace dynamic mode decomposition for stochastic Koopman analysis. *Physical Review E*, 96(033310), 2017.
- [515] F. Takens. Detecting strange attractors in turbulence. *Lecture Notes in Mathematics*, 898:366–381, 1981.
- [516] Z. Q. Tang and N. Jiang. Dynamic mode decomposition of hairpin vortices generated by a hemisphere protuberance. *Science China Physics, Mechanics and Astronomy*, 55(1):118–124, 2012.
- [517] R. Taylor, J. N. Kutz, K. Morgan, and B. Nelson. Dynamic mode decomposition for plasma diagnostics and validation. *Review of Scientific Instruments*, 89(053501), 2018.
- [518] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.
- [519] Z. Ting and J. Hui. Eeg signal processing based on proper orthogonal decomposition. In *Audio, Language and Image Processing (ICALIP), 2012 International Conference on*, pages 636–640. IEEE, 2012.
- [520] S. Tirunagari, N. Poh, K. Wells, M. Bober, I. Gorden, and D. Windridge. Movement correction in DCE-MRI through windowed and reconstruction dynamic mode decomposition. *Machine Vision and Applications*, 28(3-4):393–407, 2017.
- [521] C. Torrence and G. P. Compo. A practical guide to wavelet analysis. *Bulletin of the American Meteorological society*, 79(1):61–78, 1998.
- [522] G. Tran and R. Ward. Exact recovery of chaotic systems from highly corrupted data. *SIAM Multiscale modeling and simulation*, 15(3):1108–1129, 2017.
- [523] L. N. Trefethen. *Spectral methods in MATLAB*. SIAM, 2000.
- [524] L. N. Trefethen and D. Bau III. *Numerical linear algebra*, volume 50. Siam, 1997.
- [525] J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.
- [526] J. A. Tropp. Recovery of short, complex linear combinations via l_1 minimization. *IEEE Transactions on Information Theory*, 51(4):1568–1570, 2005.
- [527] J. A. Tropp. Algorithms for simultaneous sparse approximation. part ii: Convex relaxation. *Signal Processing*, 86(3):589–602, 2006.
- [528] J. A. Tropp. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE Transactions on Information Theory*, 52(3):1030–1051, 2006.
- [529] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, 2007.
- [530] J. A. Tropp, A. C. Gilbert, and M. J. Strauss. Algorithms for simultaneous sparse approximation. part i: Greedy pursuit. *Signal Processing*, 86(3):572–588, 2006.
- [531] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk. Beyond Nyquist: Efficient sampling of sparse bandlimited signals. *IEEE Transactions on Information Theory*, 56(1):520–544, 2010.
- [532] J. A. Tropp, A. Yurtsever, M. Udell, and V. Cevher. Randomized single-view algorithms for low-rank matrix approximation. *arXiv preprint arXiv:1609.00048*, 2016.
- [533] J. H. Tu and C. W. Rowley. An improved algorithm for balanced POD through an analytic treatment of impulse response tails. *J. Comp. Phys.*, 231(16):5317–5333, 2012.

- [534] J. H. Tu, C. W. Rowley, E. Aram, and R. Mittal. Koopman spectral analysis of separated flow over a finite-thickness flat plate with elliptical leading edge. *AIAA Paper 2011*, 2864, 2011.
- [535] J. H. Tu, C. W. Rowley, J. N. Kutz, and J. K. Shang. Spectral analysis of fluid flows using sub-Nyquist-rate PIV data. *Experiments in Fluids*, 55(9):1–13, 2014.
- [536] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz. On dynamic mode decomposition: theory and applications. *J. Comp. Dyn.*, 1(2):391–421, 2014.
- [537] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
- [538] R. Van Der Merwe. *Sigma-point Kalman filters for probabilistic inference in dynamic state-space models*. 2004.
- [539] C. Van Loan. *Computational frameworks for the fast Fourier transform*. SIAM, 1992.
- [540] D. Venturi and G. E. Karniadakis. Gappy data and reconstruction procedures for flow past a cylinder. *Journal of Fluid Mechanics*, 519:315–336, 2004.
- [541] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.
- [542] S. Volkwein. Model reduction using proper orthogonal decomposition. *Lecture Notes, Institute of Mathematics and Scientific Computing, University of Graz*. see <http://www.uni-graz.at/imawww/volkwein/POD.pdf>, 1025, 2011.
- [543] S. Volkwein. Proper orthogonal decomposition: Theory and reduced-order modelling. *Lecture Notes, University of Konstanz*, 4:4, 2013.
- [544] S. Voronin and P.-G. Martinsson. RSVDPACK: Subroutines for computing partial singular value decompositions via randomized sampling on single core, multi core, and GPU architectures. *arXiv preprint arXiv:1502.05366*, 2015.
- [545] A. Wang et al. An industrial strength audio search algorithm. In *Ismir*, volume 2003, pages 7–13. Washington, DC, 2003.
- [546] H. H. Wang, M. Krstić, and G. Bastin. Optimizing bioreactors by extremum seeking. *Adaptive Control and Signal Processing*, 13(8):651–669, 1999.
- [547] H. H. Wang, S. Yeung, and M. Krstić. Experimental application of extremum seeking on an axial-flow compressor. *IEEE Transactions on Control Systems Technology*, 8(2):300–309, 2000.
- [548] W. X. Wang, R. Yang, Y. C. Lai, V. Kováni, and C. Grebogi. Predicting catastrophes in nonlinear dynamical systems by compressive sensing. *Physical Review Letters*, 106:154101–1–154101–4, 2011.
- [549] Z. Wang, I. Akhtar, J. Borggaard, and T. Iliescu. Proper orthogonal decomposition closure models for turbulent flows: a numerical comparison. *Computer Methods in Applied Mechanics and Engineering*, 237:10–26, 2012.
- [550] C. Wehmeyer and F. Noé. Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. *The Journal of Chemical Physics*, 148(241703), 2018.
- [551] G. Welch and G. Bishop. An introduction to the Kalman filter, 1995.
- [552] P. Whittle. *Hypothesis testing in time series analysis*, volume 4. Almqvist & Wiksell, 1951.

- [553] O. Wiederhold, R. King, B. R. Noack, L. Neuhaus, L. Neise, W. an Enghard, and M. Swo-boda. Extensions of extremum-seeking control to improve the aerodynamic performance of axial turbomachines. In *39th AIAA Fluid Dynamics Conference*, pages 1–19, San Antonio, TX, USA, 2009. AIAA-Paper 092407.
- [554] K. Willcox. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers & fluids*, 35(2):208–226, 2006.
- [555] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, 40(11):2323–2330, 2002.
- [556] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. A data-driven approximation of the Koopman operator: extending dynamic mode decomposition. *Journal of Nonlinear Science*, 6:1307–1346, 2015.
- [557] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis. A kernel approach to data-driven Koopman spectral analysis. *Journal of Computational Dynamics*, 2(2):247–265, 2015.
- [558] D. M. Witten and R. Tibshirani. Penalized classification using Fisher’s linear discriminant. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(5):753–772, 2011.
- [559] F. Woolfe, E. Liberty, V. Rokhlin, and M. Tygert. A fast randomized algorithm for the approximation of matrices. *Journal of Applied and Computational Harmonic Analysis*, 25:335–366, 2008.
- [560] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(2):210–227, 2009.
- [561] C. J. Wu. On the convergence properties of the EM algorithm. *The Annals of statistics*, pages 95–103, 1983.
- [562] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip, et al. Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1):1–37, 2008.
- [563] H. Ye, R. J. Beamish, S. M. Glaser, S. C. Grant, C.-h. Hsieh, L. J. Richards, J. T. Schnute, and G. Sugihara. Equation-free mechanistic ecosystem forecasting using empirical dynamic modeling. *Proceedings of the National Academy of Sciences*, 112(13):E1569–E1576, 2015.
- [564] E. Yeung, S. Kundu, and N. Hudas. Learning deep neural network representations for Koopman operators of nonlinear dynamical systems. *arXiv preprint arXiv:1708.06850*, 2017.
- [565] B. Yildirim, C. Chryssostomidis, and G. Karniadakis. Efficient sensor placement for ocean measurements using low-dimensional concepts. *Ocean Modelling*, 27(3):160–173, 2009.
- [566] X. Yuan and J. Yang. Sparse and low-rank matrix decomposition via alternating direction methods. *preprint*, 12, 2009.
- [567] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 2528–2535, 2010.
- [568] C. Zhang and R. O. nez. Numerical optimization-based extremum seeking control with application to ABS design. *IEEE Transactions on Automatic Control*, 52(3):454–467, 2007.
- [569] H. Zhang, C. W. Rowley, E. A. Deem, and L. N. Cattafesta. Online dynamic mode decomposition for time-varying systems. *arXiv preprint arXiv:1707.02876*, 2017.

- [570] T. Zhang, G. Kahn, S. Levine, and P. Abbeel. Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search. In *IEEE Robotics and Automation (ICRA)*, pages 528–535, 2016.
- [571] W. Zhang, B. Wang, Z. Ye, and J. Quan. Efficient method for limit cycle flutter analysis based on nonlinear aerodynamic reduced-order models. *AIAA journal*, 50(5):1019–1028, 2012.
- [572] S. Zlobec. An explicit form of the moore-penrose inverse of an arbitrary complex matrix. *SIAM Review*, 12(1):132–134, 1970.
- [573] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.