Principal Component Analysis: A Natural Approach to Data Exploration

FELIPE L. GEWERS, Institute of Physics, University of São Paulo, São Paulo, SP, Brazil
GUSTAVO R. FERREIRA, Institute of Mathematics and Statistics, University of São Paulo, São Paulo, SP, Brazil
HENRIQUE F. DE ARRUDA, São Carlos Institute of Physics, University of São Paulo, São Carlos, SP, Brazil, Institute of Mathematics and Computer Science, University of São Paulo, São Carlos, SP, Brazil
FILIPI N. SILVA, São Carlos Institute of Physics, University of São Paulo, São Carlos, SP, Brazil, School of Informatics, Computing and Engineering, Indiana University, Bloomington, Indiana 47405, USA
CESAR H. COMIN, Department of Computer Science, Federal University of São Carlos, São Carlos, SP, Brazil
DIEGO R. AMANCIO, Institute of Mathematics and Computer Science, University of São Paulo, São Carlos, SP, Brazil
LUCIANO DA F. COSTA, São Carlos Institute of Physics, University of São Paulo, São Carlos, SP, Brazil

Principal component analysis (PCA) is often applied for analyzing data in the most diverse areas. This work reports, in an accessible and integrated manner, several theoretical and practical aspects of PCA. The basic principles underlying PCA, data standardization, possible visualizations of the PCA results, and outlier detection are subsequently addressed. Next, the potential of using PCA for dimensionality reduction is illustrated on several real-world datasets. Finally, we summarize PCA-related approaches and other dimensionality reduction techniques. All in all, the objective of this work is to assist researchers from the most diverse areas in using and interpreting PCA.

Felipe L. Gewers acknowedges CAPES (Finance Code 001) and CNPq (Grant No. 140442/2019-7). Gustavo R. Ferreira acknowledges financial support from CNPq (Grant No. 158128/2017-6). Henrique F. de Arruda acknowledges CAPES (Finance Code 001) and FAPESP for sponsorship (Grants No. 2018/10489-0 and No. 2019/16223-5) for sponsorship. Filipi N. Silva thanks FAPESP (Grants No. 2015/08003-4 and No. 2017/09280-7) for sponsorship. Cesar H. Comin thanks FAPESP (Grant No. 18/09125-4) for financial support. Diego R. Amancio acknowledges financial support from FAPESP (Grants No. 16/19069-9 and No. 17/13464-6). Luciano da F. Costa thanks CNPq (Grant No. 307333/2013-2) and NAP-PRP-USP for sponsorship. This work has been supported also by FAPESP Grants No. 11/50761-2 and No. 2015/22308-2.
Authors' addresses: F. L. Gewers, Institute of Physics, University of Sao Paulo, 187 Rua do Matão, São Paulo-Vila Universitária, SP, 05508-090, Brazil; email: felipe.gewers@gmail.com; G. R. Ferreira, Institute of Mathematics and Statistics, University of São Paulo, 1010 Rua do Matão, São Paulo-Vila Universitária, SP, 05508-090, Brazil; email: gustavo.r.f.95@gmail.com; H. F. de Arruda, FCM, Institute of Physics of São Carlos, University of São Paulo, 400 Trabalhador São-carlense Avenue, São Carlos, SP, 13566-590, Brazil; email: h.f.arruda@gmail.com; F. N. Silva, Indiana University Network Science Institute, 1001 IN-45, Bloomington, IN, 47408, United States; email: filipinascimento@gmail.com; C. H. Comin, Department of Computer Science, Federal University of São Carlos, 235 Washington Luiz Avenue, São Carlos, SP, 13565-905, Brazil; email: chcomin@gmail.com; D. R. Amancio, Institute of Mathematics and Computer Science, University of São Paulo, 400 Trabalhador São-carlense Avenue, São Carlos, SP, 13566-590, Brazil; email: diegoraphael@gmail.com; L. da F. Costa, FCM, Institute of Physics of São Carlos, University of São Paulo, 400 Trabalhador São-carlense Avenue, São Carlos, SP, 13566-590, Brazil; email: ldfcosta@gmail.com.

0360-0300/2021/05-ART70 $15.00
https://doi.org/10.1145/3447755

CCS Concepts: • **Applied computing → Physical sciences and engineering**; • **General and reference → Surveys and overviews**; • **Computing methodologies → Machine learning**;

Additional Key Words and Phrases: Statistical methods, principal component analysis, dimensionality reduction, data visualization, covariance and correlation

ACM Reference format:
Felipe L. Gewers, Gustavo R. Ferreira, Henrique F. de Arruda, Filipi N. Silva, Cesar H. Comin, Diego R. Amancio, and Luciano da F. Costa. 2021. Principal Component Analysis: A Natural Approach to Data Exploration. *ACM Comput. Surv.* 54, 4, Article 70 (May 2021), 34 pages.
https://doi.org/10.1145/3447755

"Frustra fit per plura quod potest fieri per pauciora." *William of Occam*

1 INTRODUCTION

Science has always relied on the collection, organization, and analysis of data and features. A proverbial example that promptly comes to mind is the criticality of Tycho Brahe's observations for the development of Galileo's studies on gravitation [Ferguson 2002]. Since that time, substantial technological advances, in particular, in electronics and information science, have implied an ever-increasing accumulation of large amounts of the most varied types of data, extending from eCommerce to astronomy. Not only have more types of data become available, but traditional measurements in areas such as particle physics are now performed with increased resolution and in substantially larger numbers. Such trends are now known as the *data deluge* [Bell et al. 2009]. However, such vast quantities of data are, by themselves, of no great avail unless appropriate methods are applied to identify the most relevant *information* contained in such repositories, a process known as *data mining* [Hand 2007]. Indeed, provided effective means are available for mining, particularly valuable information can be extracted. For instance, it is likely that the information in existing datasets would already be enough to allow us to make substantial medical advances. The importance of organizing and summarizing data can therefore be hardly exaggerated.

While a definitive solution to the problem of data mining remains elusive, there are some well-established approaches that have been useful for organizing and summarizing data [Bishop 2006]. Perhaps the most popular among these is ***Principal Component Analysis (PCA)*** [Abdi and Williams 2010; Costa and Cesar Jr 2009; Jolliffe 1986; Pearson 1901]. Being such a popular method, several works describing different aspects of PCA are available; see, for instance, Brereton [2018]; Bro and Smilde [2014]; Esbensen and Geladi [2009]; Tharwat [2016]; Wold et al. [1987].

Let us organize the several (n) features of each object or individual i in terms of a respective *feature vector* $\mathbf{x}_i'$, existing in an n-dimensional *feature space*. PCA can then be understood as a statistical method in which the coordinate axes of the feature space are rotated so that the first axis coincides with the direction of maximum possible data dispersion (as quantified by the variance), the second axis with the direction of second maximum dispersion, and so on. This principle is illustrated with respect to a simple situation with $n = 2$ in Figure 1. Here, we have $p = 64$ objects (real beans), each described by $n = 2$ respective features, which are themselves organized as a respective feature vector. More specifically, each object i has respective measurements x_{i1} (diameter) and x_{i2} (square root of the bean area).

When mapped into the respective two-dimensional feature space, these objects define a distribution of points that, in the case of this example, assumes an elongated shape. Finding this type of point distribution in the feature space can be understood as indications of *correlation* between the features. In the case of beans, their shape can be approximated as a disk, for which the area is given as pi times square radius (equal to half the diameter). Thus, except for shape variations, the

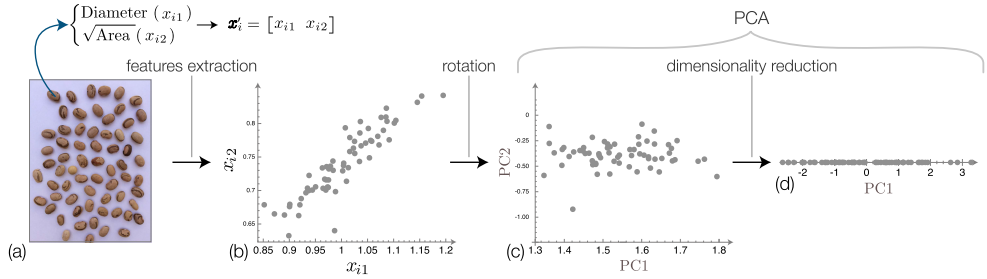

Fig. 1. PCA example on a real-world situation. Each bean (a) is characterized in terms of two features: diameter x_{i1} and square root of area x_{i2}. Though these two features are intrinsically related in a direct fashion, bean shape variations induce a dispersion of the objects when mapped into the features space (b). PCA allows the identification of the orientation of maximum data dispersion (c). As the dispersion in the resulting second axis is relatively small, this axis can be eventually disregarded (d).

two chosen features are directly related and would be, in principle, redundant. However, because no two beans have exactly the same shape, we have the dispersion observed in the feature space (Figure 1(b)).

The application of PCA to this dataset will rotate the coordinate system, yielding the new axes identified as PC1 and PC2 in Figure 1(c). The maximum data dispersion in one dimension is now found along the first axis, PC1. The second axis, PC2, will be characterized by the second largest one-dimensional dispersion. The coordinate values obtained after the rotation are sometimes called *principal component scores*.

Interestingly, provided the original data distribution is elongated enough, it is now possible to discard the second axis without great loss of overall data variation. The resulting feature space now has dimension $\ell = 1$ (Figure 1(d)).

The *essence* of PCA applications, therefore, consists in simplifying the original data with minimal loss of overall dispersion, paving the way to a reduction of the dimensionality with which the data is represented. Typical applications of PCA are characterized by having $\ell << n$. Observe that PCA ensures maximum dispersion projections and promotes dimensionality reduction, but does not guarantee that the main axes (along the directions of largest variation in the original data) will necessarily correspond to the directions that would be more useful for each particular study. For instance, if one is aiming at separating categories of data, the direction of best discrimination may not necessarily correspond to that of maximum dispersion, as provided by PCA. Indeed, a more robust approach to exploring and modeling a dataset could involve, in addition to PCA, the application of several types of projections, including: **Linear Discriminant Analysis (LDA)**, **Independent Component Analysis (ICA)**, maximum entropy, among many others [Cunningham 2008; Fodor 2002]. However, this approach can imply in substantial computation cost because of the nonlinear optimization required by many of the aforementioned projections. Interestingly, in case of datasets characterized by the presence of correlations between the features, PCA can still be applied prior to the other computationally more expensive projections to obtain data simplification, therefore reducing the overall execution time and catering for more significant statistics. Thus, one particularly important issue with PCA is its efficiency for simplifying, through decorrelation, datasets typically found in the real-world or simulations.

In the example above, we have $n = 2$, and it is simple to visualize the data as a scatter plot. Since, in general, n is larger than 2 or 3, it becomes a challenge to visualize the overall distribution of samples in a typical feature space. Thus, by projecting the data into 2 or 3 dimensions, PCA is also a powerful method allowing insightful *visualizations* of high-dimensional data.

Here, the concept of PCA as well as several issues regarding its practical applications are presented in an intuitive and accessible manner. Special efforts have been invested to achieve a work that could be interesting to researchers from diverse levels of expertise and areas, including a step-by-step presentation of PCA as well as a complete application example. More advanced issues such as proof of dispersion maximization, stability of the covariance matrix, and so on, are provided as supplementary material that will probably be of interest to more experienced readers as well.

This work is organized as follows. In Section 2, we introduce the basic statistical concepts underlying PCA. Then, we explain the mathematical underpinnings of PCA in Section 3 in a more formal setting, and address the issue of standardization in Section 4. Section 5 is dedicated to explaining the biplot technique, which is a very popular approach to interpreting a PCA. We discuss the use of PCA for outlier detection in Section 6. Sections 7 and 8 are dedicated to applications of PCA to real-world data. In the former, we present a step-by-step application of PCA to a real dataset of bean pictures, and in the latter, we illustrate the results of applying PCA to datasets from different fields, paying particular attention to the variance distribution along the axes. Then, in Sections 9 and 10, we outline other approaches to dimensionality reduction; Section 9 focuses on variants of PCA itself, while Section 10 briefly introduces other dimensionality reduction methods based on more sophisticated concepts.

2 CORRELATION, COVARIANCE, AND PEARSON CORRELATION

In this section, we present some basic statistical concepts that are used for estimating the relationship between variables. These concepts are intrinsically related to PCA, since they provide the basis for its interpretation.

In practice, the analysis of data is typically performed in terms of observations or features of some entities, along a *sampling*/*observation* process. For instance, in the beans case example in the previous section, we started with an image of real beans, from which two features were estimated and used for identifying the main orientations of data variation by using PCA. From a theoretical point of view, it is beneficial to abstract such data as a statistical model. This section develops the basic concepts leading to PCA, in particular, the idea of probability density and respective moments, allowing us to understand PCA as a statistical transformation.

The basic concept in statistical modeling is of the idea of a *random variable*. In principle, every feature in the real-world corresponds to the sampling (or observation) of a respective random variable. In the case of the beans example, measuring the area of a particular bean can be related to sampling a random variable associated to a set of beans.

These variables, which correspond to an abstract model of the feature and are typically represented by capital calligraphic letters, such as $\mathscr{X}$, can be characterized in terms of the respective probability density function $p(X)$. These functions are special in the sense that they provide all information that is possible to be obtained about a random variable. A probability density function must be larger than 0 for all x and the integration of the function over the entire real line must be equal to one. Given such a function, it is possible to obtain the probability of observing values of $\mathscr{X}$ within an interval $a \leq \mathscr{X} \leq b$ as follows:

$$P(a \leq \mathscr{X} \leq b) = \int_a^b p(x)\, dx. \tag{1}$$

The probability density function can also be mapped into scalar values, more specifically moments, that represent specific overall properties of the original model. For instance, the *average* of $\mathscr{X}$, more formally called *expectation*, can be defined as

$$\bar{\mathscr{X}} = E[\mathscr{X}] = \int_{-\infty}^{\infty} xp(x)\, dx. \tag{2}$$

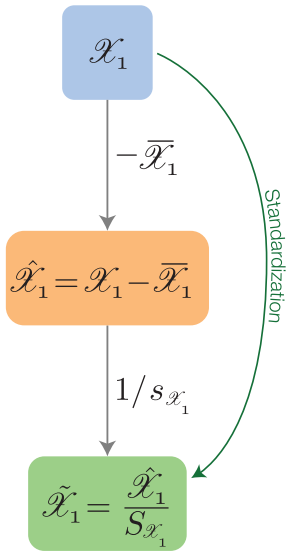

Fig. 2. An original random variable $\mathscr{X}$ can be statistically transformed to the coordinates origin by subtracting its respective expectation, yielding the new random variable $\hat{\mathscr{X}}$. A subsequent division by the standard deviation will normalize the variable's dispersion, yielding a dimensionless variable $\tilde{\mathscr{X}}$.

Furthermore, the *standard deviation* of $\mathscr{X}$ can be defined as

$$S_{\mathscr{X}} = \sqrt{E[\mathscr{X}^2] - (E[\mathscr{X}])^2}. \tag{3}$$

Given a random variable $\mathscr{X}$, any function that maps it into another random variable can be understood as a statistical transformation [Feller 2008]. Figure 2 illustrates two particularly important such transformations, corresponding to translation to the coordinate origin (by subtracting the respective expectation $E[\mathscr{X}]$), also called *centralization*, and subsequent variance normalization (dividing by the respective standard deviation $S_{\mathscr{X}}$). The combined application of these two transformations yields the well-known operation of *standardization* [Everitt and Skrondal 2002]. As often adopted, we will use the term *normalization* to refer to any generic alterations of the original features aimed at making them more compatible, reserving the term *standardization* to the specific statistical transformation involving subtraction of the average and subsequent division by the standard deviation.

After translating to the coordinate origin, the new random variable $\hat{\mathscr{X}}$ will have zero expectation. After a random variable $\mathscr{X}$ is standardized into $\tilde{\mathscr{X}}$, this new variable will necessarily have zero expectation and unit standard deviation (and, thus, unit variance). Additionally, most of the random variable observations will be comprised in the interval ranging between -2 and 2 due to Chebyshev's inequality [Bertsekas and Tsitsiklis 2002].

Given two random variables $\mathscr{X}_1$ and $\mathscr{X}_2$, it is important to consider statistical features of their possible relationship or joint variation. Such features can then be used to quantify how much two variables are related, an aspect that is directly associated to *data redundancy*. There are three main basic ways to do so: *expectation of the product*, *covariance*, and *(Pearson) coefficient of correlation* [Pearson 1895]. The expectation of the products between $\mathscr{X}_1$ and $\mathscr{X}_2$ is calculated as

$$E[\mathscr{X}_1 \mathscr{X}_2]. \tag{4}$$

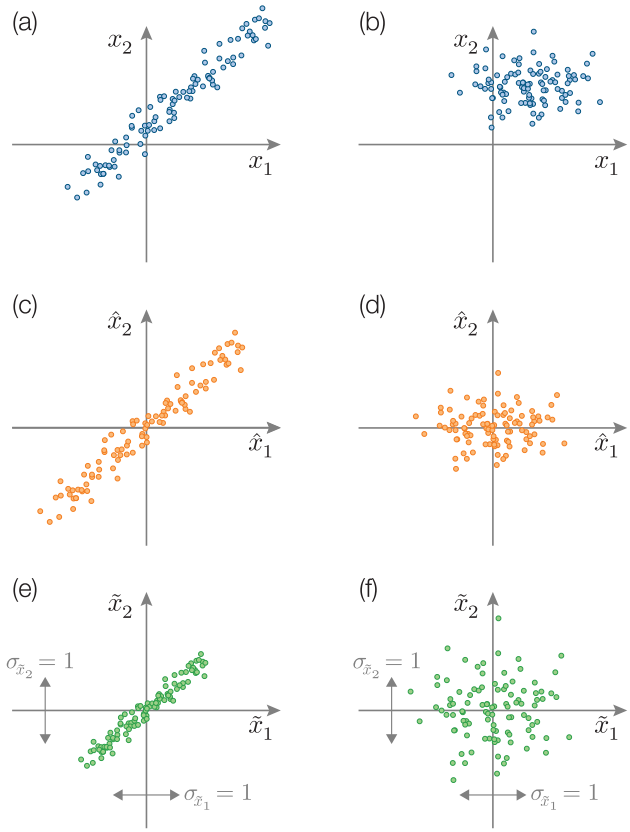

Fig. 3. Example of data standardization. (a, b) Original data $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, which can be considered as samples of random variables $\mathscr{X}_1$ and $\mathscr{X}_2$. (c, d) Zero-centered data after subtracting the expected values. (e, f) Unit-variance data after dividing by their respective standard deviations.

This quantity already expresses some level of relationship between the two variables. Consider the example in Figure 3(a), which shows possible samples of the two variables $\mathscr{X}_1$ and $\mathscr{X}_2$, respectively, represented as $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. This figure shows that $\mathscr{X}_1$ has an evident relationship with $\mathscr{X}_2$. Most of the products between the sampled values for these two variables are positive, indicating positive $E[\mathscr{X}_1\mathscr{X}_2]$. Consider now the distribution in Figure 3(b). It can be easily verified that a positive value $E[\mathscr{X}_1\mathscr{X}_2]$ will again be obtained, expressing a relationship between $\mathscr{X}_1$ and $\mathscr{X}_2$, which is indeed true in the sense that *both* these variables tend to have relatively large, positive values.

Another statistical feature of relationships between two random variables is the *covariance*. As hinted in its own name, this feature quantifies joint variations between the two variables. Mathematically, the covariance between $\mathscr{X}_1$ and $\mathscr{X}_2$ is given as [Everitt and Skrondal 2002]

$$\begin{aligned} K_{12} = \mathrm{Cov}(\mathscr{X}_1, \mathscr{X}_2) &= E[\hat{\mathscr{X}_1}\hat{\mathscr{X}_2}] \\ &= E[(\mathscr{X}_1 - E[\mathscr{X}_1])(\mathscr{X}_2 - E[\mathscr{X}_2])]. \end{aligned} \tag{5}$$

Thus, the covariance between $\mathscr{X}_1$ and $\mathscr{X}_2$ corresponds to the expectation of the product between variables $\hat{\mathscr{X}_1}$ and $\hat{\mathscr{X}_2}$. Figures 3(c) and 3(d) show the effect of moving the original data distributions in Figures 3(a) and 3(b), respectively, to the coordinates origin. Observe that the random variables $\hat{\mathscr{X}_1} = \mathscr{X}_1 - E[\mathscr{X}_1]$ and $\hat{\mathscr{X}_2} = \mathscr{X}_2 - E[\mathscr{X}_2]$ have an expected value of zero. The products between

the variables tend to be positive in the case of Figure 3(c), indicating that $\mathscr{X}_1$ and $\mathscr{X}_2$ present a joint tendency to vary together. However, in the case of the point distribution in Figure 3(d), the products tend to cancel between the positive values obtained for the quadrants 1 and 3 and the negative values in the quadrants 2 and 4, resulting in nearly null overall covariance between $\mathscr{X}_1$ and $\mathscr{X}_2$. Observe that the point distribution in Figure 3(b) therefore yields a positive expectation of the product, but nearly null covariance, while positive expectation of the product and covariance are obtained for the points in Figure 3(a). When two variables $\mathscr{X}_1$ and $\mathscr{X}_2$ have a null covariance value, they are said to be *uncorrelated*.

Now, we proceed to the (Pearson) coefficient of correlation between $\mathscr{X}_1$ and $\mathscr{X}_2$. This quantity is defined as

$$\begin{aligned} \text{PCorr}(\mathscr{X}_1, \mathscr{X}_2) &= \text{Cov}(\hat{\mathscr{X}_1}/S_{\mathscr{X}_1}, \hat{\mathscr{X}_2}/S_{\mathscr{X}_2}) && (6)\\ &= E\left[\frac{(\mathscr{X}_1 - E[\mathscr{X}_1])(\mathscr{X}_2 - E[\mathscr{X}_2])}{S_{\mathscr{X}_1} S_{\mathscr{X}_2}}\right]. && (7) \end{aligned}$$

The coefficient of correlation between $\mathscr{X}_1$ and $\mathscr{X}_2$ therefore corresponds to the expectation of the product between $\tilde{\mathscr{X}_1}$ and $\tilde{\mathscr{X}_2}$, or the covariance between $\hat{\mathscr{X}_1}/S_{\mathscr{X}_1}$ and $\hat{\mathscr{X}_2}/S_{\mathscr{X}_2}$. Figures 3(e) and 3(f) depict the distributions of points in Figure 3(a) and 3(b) after translation to the coordinates origin and division by the variables standard deviations. The yielded standardized variables $\tilde{\mathscr{X}_1}$ and $\tilde{\mathscr{X}_2}$ are dimensionless and both have unit variance and standard deviation. This implies the orientation of the main elongation in Figure 3(a) to change. The Pearson correlation coefficient for the point distributions in Figures 3(e) and 3(f) can be immediately estimated in terms of the expectation of the products $\tilde{\mathscr{X}_1}, \tilde{\mathscr{X}_2}$, which are positive for Figure 3(e) and nearly null for Figure 3(f). The Pearson correlation coefficient always lies in the range $-1 \leq \text{PCorr}(\mathscr{X}_1, \mathscr{X}_2) \leq 1$. In case $\text{PCorr}(\mathscr{X}_1, \mathscr{X}_2) = 1$ or $\text{PCorr}(\mathscr{X}_1, \mathscr{X}_2) = -1$, the two random variables are perfectly related by a straight line and are, consequently, totally redundant one another. Indeed, the closer the absolute value of $\text{PCorr}(\mathscr{X}_1, \mathscr{X}_2)$ is to one, the more redundant one of the variables is with the other. The two variables will also be redundant for *relatively* larger values of $\text{Cov}(\mathscr{X}_1, \mathscr{X}_2)$, but in a non-normalized way. Figure 4 summarizes the relationships between these features.

It should be observed that the three statistical joint features discussed in this section assume *linear* relationship between pairs of random variables. Other features can be used to characterize nonlinear relationships, such as the *Spearman's rank correlation* and *mutual information* [Hair et al. 1998].

In a problem involving $n \geq 1$ random variables, the variables can be organized as a row *random vector* $\boldsymbol{\mathscr{X}} = (\mathscr{X}_1, \mathscr{X}_2, \ldots, \mathscr{X}_n)$. For $n > 1$, it is possible to calculate the covariance or Pearson correlation coefficient for all pairs of random variables. It is possible to transform random vectors in several ways. One special type of a random vector transformation corresponds to a linear transformation, which can be expressed as

$$\boldsymbol{\mathscr{Y}} = \boldsymbol{\mathscr{X}} A. \tag{8}$$

PCA can be understood as a linear transformation of the original variables $\boldsymbol{\mathscr{X}}$ that yields new random variables so that the variation of the original data is maximized along the first of these variables. Another consequence of this transformation is that the new random variables are completely uncorrelated. In the next section, we formally explain the PCA calculation and related properties.

3 PRINCIPAL COMPONENT ANALYSIS EXPLAINED

In this section, we present the mathematical formulation of PCA. For simplicity's sake, we develop this formulation by integrating the conceptual framework presented in the introduction with the

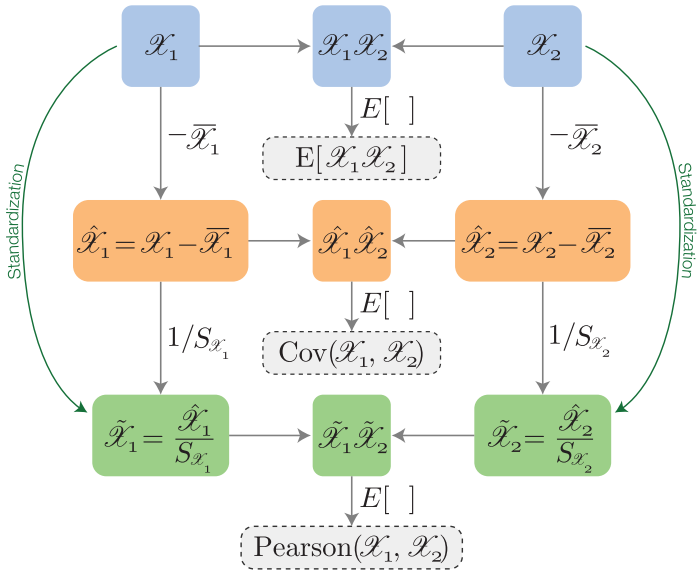

Fig. 4. The covariance and Pearson correlation coefficient between two given random variables $\mathscr{X}_1$ and $\mathscr{X}_2$ can be understood as the expectation $E[\ ]$ of the products between, respectively: the two original variables translated to the coordinates origin, and the two original variables moved to the origin and normalized by the respective standard deviations.

Table 1. Typical Organization of the Input Data for PCA

	Feature 1	Feature 2	...	Feature n
Object 1	5.4	2.4	...	12.3
Object 2	7.5	3.5	...	10.3
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
Object p	8.3	1.4	...	14.2

The matrix containing the values of all objects for all features is called the data matrix.

basic statistical concepts covered in Section 2. Also, in Appendix B, we provide some basic mathematical concepts complementing the calculations presented in this section.

3.1 Algebraic Viewpoint

Consider that we are interested in n different variables (or features), conceptually represented by the random variables $\mathscr{X}_1, \mathscr{X}_2, \ldots, \mathscr{X}_n$, which can be further combined in the random vector $\boldsymbol{\mathscr{X}} = (\mathscr{X}_1, \mathscr{X}_2, \ldots, \mathscr{X}_n)$ in a feature space of dimension n. Through observations and measurements, we obtain a *sample* of our random vector, i.e., a set of p row vectors $\boldsymbol{x}'_1, \boldsymbol{x}'_2, \ldots, \boldsymbol{x}'_p$. These data are then summarized in the data matrix X with dimension $p \times n$, where each of the p lines represents an object/individual, and each column j, $1 \leq j \leq n$, expresses a respective feature/variable $\boldsymbol{x}_j$. Also, the values measured for the ith object (its feature vector) are represented as $\boldsymbol{x}'_i$. Therefore, each element x_{ij} of X corresponds to the of the jth feature for the ith object of our sample, Table 1 shows an example of a data matrix.

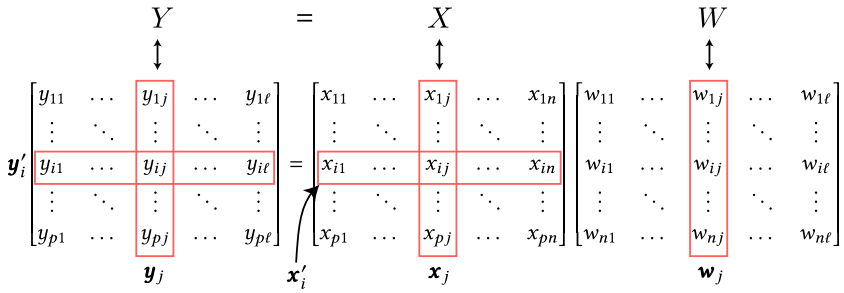

Fig. 5. The basic components in the PCA transformation (Equation (9)). The original data is organized as matrix X, each column of which corresponds to a feature $\boldsymbol{x}_j$. Also, each object is characterized by a vector $\boldsymbol{x}_i'$ in the rows of X. The transformation matrix, W, has each column corresponding to an eigenvector $\boldsymbol{w}_j$ of the data covariance or correlation matrix. The jth column of the resulting matrix Y contains the scores of the jth principal component, represented as $\boldsymbol{y}_j$, and each projected object i is characterized by the row vector $\boldsymbol{y}_i'$.

Being a linear combination, PCA can be expressed in the following matrix form (as shown in a more conceptual level at the end of the previous section):

$$Y = XW. \tag{9}$$

In other words, the PCA transformation corresponds to multiplying the original features X by an appropriate transformation matrix W. Figure 5 shows the notation used for representing each of the involved variables. To perform the PCA of a given dataset, the transformation matrix W is obtained and then Equation (9) is applied. The derivation of W is described as follows.

First, to translate our variables to the coordinate origin, we estimate the average of each variable as

$$\mu_{\boldsymbol{x}_j} = \frac{1}{p}\sum_{i=1}^{p} x_{ij}. \tag{10}$$

The features are then brought to the coordinate origin by subtracting the respective averages, i.e.,

$$\hat{\boldsymbol{x}}_j = \boldsymbol{x}_j - \mu_{\boldsymbol{x}_j}. \tag{11}$$

The new data matrix containing all the variables $\hat{\boldsymbol{x}}_j$ in its columns is henceforth represented as $\hat{X}$. The covariance matrix of the data matrix X can now be defined as

$$K = Cov(X) = \frac{1}{p-1}\hat{X}^T\hat{X}. \tag{12}$$

PCA is often performed using the standardized covariance matrix, the Pearson correlation matrix, instead of the covariance matrix. A discussion about the two approaches is developed in Section 4. In this section, we present the PCA transformation by using the covariance matrix. However, for the Pearson correlation matrix, the procedure is analogous.

At this stage, we have estimated the covariance matrix of the original features. The next step consists of obtaining the necessarily non-negative eigenvalues λ_j and their respective eigenvectors $\boldsymbol{w}_j$, $1 \le j \le n$, of K. These eigenvalues are then sorted in decreasing order. This step can be calculated through eigendecomposition by solving the linear equation

$$K\boldsymbol{w}_j = \lambda_j \boldsymbol{w}_j; \tag{13}$$

more details about this kind of equation can be found in Appendix B. Optimized libraries for calculating eigenvalues and eigenvectors can be found in most of programming languages. Such

libraries usually return eigenvectors having unitary norm, that is, the eigenvectors have the property

$$\sum_{i=1}^{n} w_{ij}^2 = 1, \tag{14}$$

where w_{ij} is the ith element of eigenvector $\boldsymbol{w}_j$. This property is important to ensure that the transformation strictly represents a rotation of the original data, thus avoiding any rescaling of the values. Some libraries may not return normalized eigenvectors. In such cases, the eigenvectors can be normalized by dividing each eigenvector $\boldsymbol{w}_j$ by its respective norm.

The eigenvectors are then combined in columns, in decreasing order of their eigenvalues, to obtain the transformation matrix, i.e.,

$$W = \begin{bmatrix} \uparrow & & \uparrow \\ \boldsymbol{w}_1 & \cdots & w_n \\ \downarrow & & \downarrow \end{bmatrix}. \tag{15}$$

So, all we need to do now to obtain the PCA projection of a given object i is to use Equation (9), i.e.,

$$\boldsymbol{y}'_i = \boldsymbol{x}'_i W, \tag{16}$$

where $\boldsymbol{x}'_i$ and $\boldsymbol{y}'_i$ represent the feature vectors of object i in the original and transformed spaces, respectively. Each entry y_{ij} of the post-PCA projection data matrix is called a *principal component score* of the matrix Y. Each new feature $\boldsymbol{y}'_i$ is defined by a linear combination of the original features in which the weights are given by the eigenvectors $\boldsymbol{w}_j$.

An important point to be kept in mind is that each dataset yields its own transformation matrix W. In other words, this matrix *adapts* to the data to provide some critically important properties of PCA, such as the ability to completely decorrelate the original variables and to concentrate variation in the first PCA axes. Each of the new axes is parallel to an eigenvector of W and is also called a *principal component*.

The eigenvectors $\boldsymbol{w}_j$ are normalized, and thus indicate only the direction of the principal components. Another approach often used to interpret and analyze a PCA transformation is to define the *loadings vectors*, calculated as

$$\boldsymbol{f}_j = \sqrt{\lambda_j}\boldsymbol{w}_j, \tag{17}$$

which, beyond the principal component's direction, also contains information about the explained variance along that direction (see Section 5 for additional details).

Regarding more practical aspects, the pseudocode and respective implementations of PCA in the R, Python and Scilab programming languages are included in Section 1 of the Supplementary Information. A non-commercial software for calculating PCA is provided in Weka.[1]

The calculation of the principal components using the covariance matrix is intuitive, but in some specific situations it might be numerically unstable; and if n is large, it can also be computationally expensive, since it involves the calculation of an $n \times n$ matrix and the determination of its eigenvalues and eigenvectors. Therefore, many software and libraries implement PCA using a methodology called **Singular Value Decomposition (SVD)**. In Section 2 of the Supplementary Information, we provide a brief discussion regarding SVD and its relationship with PCA.

[1] https://www.cs.waikato.ac.nz/ml/weka/.

3.2 Dimensionality Reduction

So far, the transformed data matrix Y still has the same size as the original data matrix X. That is to say, the transformation implied by Equation (9) only remapped the data into a new feature space, defined by the eigenvectors of the covariance matrix. This process can be understood as a rotation of the coordinate system that aligns the axes along with the most extensive data variation directions. Decreasing the number of variables corresponds to keeping the first $\ell \leq n$ principal components so that W is reduced to an $n \times \ell$ matrix. So, the new feature vectors $\boldsymbol{y}'_j$ have only ℓ elements (or dimensions). Equivalently, we are keeping only the transformed vectors $\boldsymbol{y}_1$ to $\boldsymbol{y}_\ell$ and discarding the rest.

Due to orthogonality of W (see Section 3 of the Supplementary Information), an important characteristic of PCA is that the *total data variance* is preserved under rotation of the axes, and, therefore, also by the PCA transformation (Equation (9)):

$$TV = \sum_{j=1}^{n} \sigma^2_{\boldsymbol{x}_j} = \sum_{j=1}^{n} \sigma^2_{\boldsymbol{y}_j}. \tag{18}$$

In other words, the total variance of the original data is identical to that of the new data produced by rotating the variables $\boldsymbol{x}_j$ according to the matrix W. It allows us to define, for each principal component, an *explained variance* $EV_j = \sigma^2_{\boldsymbol{y}_j}$ that quantifies the amount of the total variance held by that component.

Now, when choosing a value of ℓ, we are discarding part of the data's variance, and keeping only the part contained along with the directions $\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_\ell$, corresponding to the new variables $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_\ell$. Under the discussion above, the *cumulative variance* after the projection is

$$CV = \sum_{j=1}^{\ell} EV_j = \sum_{j=1}^{\ell} \sigma^2_{\boldsymbol{y}_j}, \tag{19}$$

which can be also expressed in terms of the ratio

$$G = (100\%)\,\frac{CV}{TV}. \tag{20}$$

An interesting aspect of *PCA* is that CV can be calculated in a simpler manner, as it can be shown that the variance along with the jth principal component is equivalent to the correspondent eigenvalue of the covariance matrix K:

$$\sigma^2_{\boldsymbol{y}_j} = \lambda_j. \tag{21}$$

Therefore, CV becomes the sum of the ℓ largest eigenvalues of K. Furthermore, the eigenvalues' decreasing order implies that the first principal components have the largest explained variances. Finally, the number ℓ of variables can be defined in relation to G. For instance, if we desire to preserve 70% of the overall variance after PCA, then we choose ℓ so that we have $G \approx 70\%$.

Several methods have been defined to assist in the choice of a suitable ℓ. For instance, *Tipping and Bishop* [Tipping and Bishop 1999] defined a probabilistic version of PCA based on a latent variable model. This approach enables an effective dimensionality reduction of the dataset by using a Bayesian treatment of PCA [Bishop 1999]. In Hansen et al. [1999], a generalization error was employed to select the number of principal components, which was evaluated analytically and empirically. To compute this error analytically, the authors modeled data using a multivariate normal distribution.

In principle, there is no guarantee that an $\ell < n$ exists ensuring that a given variance will be achieved. It depends drastically on the distribution of the λ_j values, which, in turn, depend on the

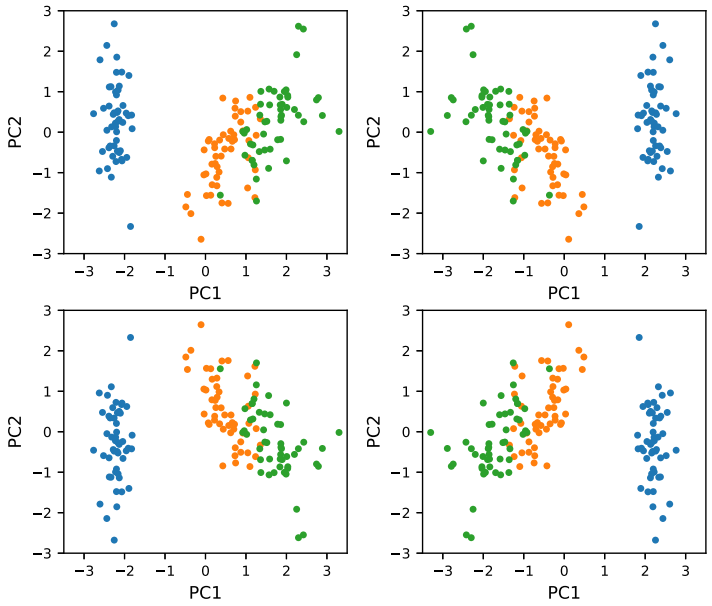

Fig. 6. Four two-dimensional PCA projections can be obtained for the Iris dataset or, indeed, any other data. The colors identify the three categories in the Iris dataset and are included here only for reference, being immaterial to PCA.

structure of the dataset. More specifically, datasets with highly correlated variables will allow for higher variance preservation in the first few principal components.

Finally, since the $\boldsymbol{y}_j$ are a linear combination of the $\boldsymbol{x}_j$, PCA rarely results in the removal of any of the original features, since all of the $\boldsymbol{x}_j$ can contribute to the eigenvectors of W. For a variant of PCA that does discard some of the original features, see Sparse PCA in Section 9.1.

3.3 Other Aspects of PCA

It may come as a surprise to know that any of the PCA axes' directions are not determined, which follows immediately from the fact that if $\boldsymbol{w}_j$ is an eigenvector of a matrix K, so is $-\boldsymbol{w}_j$. This property implies that any of the principal components can have its orientation reversed without incurring in any error. In other words, the orientations of the PCs are *arbitrarily* defined. Figure 6 illustrates this interesting and important property of PCA. This figure shows the four possible PCA projections of the Iris dataset [Fisher 1936] into two dimensions. The same is true for any dataset. Any of the PCA diagrams in this figure are correct and are, indeed, equivalent to one another.

As aforementioned, the PCA transformation is a rotation of the axes in the feature space. In Section 3 of the Supplementary Information, we demonstrate that any rotation preserves the total data variance, and the post-PCA covariance matrix $Cov(Y)$ is a diagonal matrix having the eigenvalues of $Cov(X)$ in descending order. The fact that $Cov(Y)$ is diagonal implies that the post-PCA data matrix Y is completely decorrelated. Furthermore, in Section 4 of the Supplementary Information, we prove that the principal components are aligned to the directions of maximal variance, in decreasing order.

Given these characteristics, it is interesting to observe that the efficacy of PCA in explaining variance is, to a good extent, a consequence of two properties of the eigenvectors associated to each principal component. First, we have that these eigenvectors are orthogonal (a consequence of the covariance matrix being symmetric). Second, we also have that each eigenvector corresponds to a "prototype" of the data, in the sense of having a significant similarity with the original data. Thus,

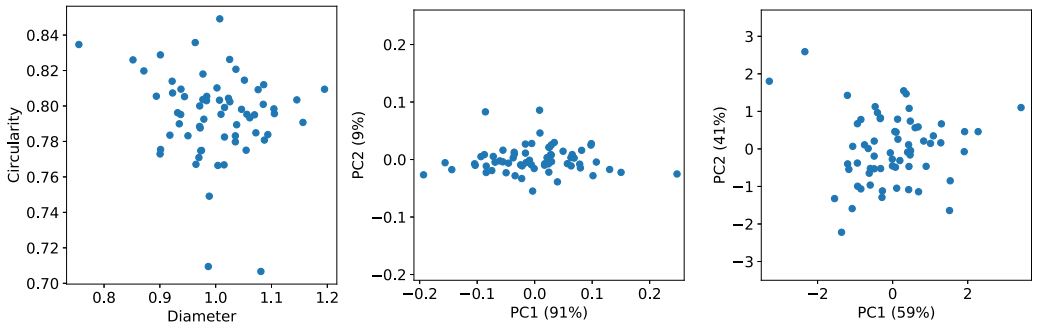

Fig. 7. Influence of standardization for a dataset composed of the diameter and circularity of the beans discussed in Section 1. (a) Scatter-plot between the diameter and circularity. (b) PCA result when applied to the unnormalized features. (c) PCA result after normalizing the features to have zero mean and unit variance.

if a given feature vector is roughly aligned with one of the eigenvectors, then it will necessarily differ from the other eigenvectors as a consequence of the orthogonality between eigenvectors.

4 TO STANDARDIZE OR NOT TO STANDARDIZE?

We have already seen that random variables can, through statistical transformations, be normalized in several ways to address specific requirements. The application of PCA often implies the question of whether or not to normalize the original data. Quite often, the dataset is *standardized* prior to PCA [Jolliffe 1986], but other normalizations can also be considered. In this section, we discuss the important issue of data standardization prior to PCA.

A possible way to address this issue is to first consider the respective implications. As seen in Section 2, data standardization of a random variable (or vector) leads to corresponding dimensionless new variables that have zero mean and unit variance (i.e., similar scales). Therefore, *all* standardized, dimensionless variables will have *similar* ranges of variation. So, standardization can be particularly advisable as a way to avoid the overdue influence of the magnitude of certain variables when the original features have significantly different dispersion or scales. When the original features already have similar dispersion, standardization has a smaller effect.

To better understand the influence of standardization on PCA, let us go back to the beans dataset discussed in the Introduction. Besides the area and diameter, another possible feature to characterize the beans is the circularity, which quantifies how circular a given bean is. Larger circularity values mean that the shape of the bean is more circular. Figure 7(a) shows a scatter plot between the diameter and circularity of the beans. It is clear that these two quantities have low covariance and Pearson correlation, that is, more circular beans are not necessarily larger or smaller than more elongated ones, and vice versa. Therefore, it is expected that the application of PCA to these two features will result in two principal components having similar explained variances, since the diameter and circularity seems to provide complementary information.

Figure 7(b) shows the result of applying PCA to the beans' diameter and circularity. The explained variance of each principal component is also indicated in the axes labels. Interestingly, the first principal component resulted in an explained variance of 91%, suggesting that most of the data can be represented by a single variable. This happened because the variance of the diameter is much larger than the variance of the circularity (about 10 times larger; notice the difference in scaling in the axes of Figure 7(a)). Therefore, the diameter accounted for most of the variance observed in the first principal component.

A possible approach to avoid the variance of one feature dominating the others in PCA is to standardize the data so that each feature has zero mean and unit variance. This is equivalent to

calculating the principal components using the Pearson correlation matrix instead of the covariance matrix. Figure 7(c) shows the result of applying PCA to the standardized data. Now, the first principal component accounts for 59% of the explained variance, while the explained variance of the second is 41%. Thus, PCA indicates that two features are needed to represent the data.

Considering the result for the standardized data, one may conclude that a good rule of thumb is that the features should be standardized before applying PCA, except in situations where their variances are of similar scales. However, standardizing the features might also lead to wrong conclusions about the data. For instance, the circularity, by definition, has values between 0 (a line) and 1 (a circle). Figure 7(a) shows that circularity varies little among the beans, since most of the values are in the range $[0.76, 0.82]$. It could be the case that the circularity is actually identical among the considered beans, and the observed variation is caused by some error in the procedure used to measure circularity, such as noise in the image (in case the values are measured using Digital Image Processing methods) or insufficient resolution of the measuring apparatus. In such cases, the variation of the circularity would not contain any particularly important information about the beans, but the standardization procedure would lead to two principal components having similar explained variances. This effect will be even more pronounced for datasets containing hundreds of features. If the variance of some of the features is uninformative, then standardization might artificially amplify their significance in PCA.

Thus, if the variation of a feature is intrinsic to the data (i.e., it is not an artifact) and meaningful, then standardization can be used to make the information provided by the feature as relevant as the other considered features. However, if the variation is a consequence of an unwanted effect (e.g., experimental error or noise), then standardization may emphasize what should have been otherwise eliminated. In such cases, either the noise should be reduced by some means, or standardization avoided. Section 6 of the Supplementary Information contains additional discussion about the influence of unwanted variations of a feature in PCA.

5 PCA LOADINGS AND BIPLOTS

The principal axes identified by PCA are linear combinations of the original features. As such, an interesting question arises regarding the identification of how those features are related to the implemented projection. For instance, in the case of the beans example in Section 1, we have that the first principal component is defined as $\boldsymbol{y}_1 = 0.82\boldsymbol{x}_1 + 0.57\boldsymbol{x}_2$, where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ represent, respectively, the diameter and square root of the area of the beans. In other words, we have that the *weights* of the *Diameter* and $\sqrt{Area}$ features are 0.82 and 0.57, respectively. As a consequence, the two original features contribute almost equally to the first principal component.

A possible manner to visualize the relationship between the original and principal components consists of projecting the former into the obtained PCA space. Figure 8(a) illustrates such a projection with respect to the Iris dataset [Fisher 1936]. This dataset involves four features for each individual (each individual being a flower of the genus *Iris*), namely: sepal length, sepal width, petal length, and petal width. The projections of the axes corresponding to each of these four original variables are identified by the four respective vectors in the PCA space in Figure 8(a). Each of these projected vectors is obtained by multiplying the eigenvectors by the respective unit vector associated with the feature. For instance, in the case of the sepal length variable, the projected vector is calculated as

$$\left[\text{Sepal length}_{PC_1}\ \text{Sepal length}_{PC_2}\right] = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix} \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ w_{31} & w_{32} \\ w_{41} & w_{42} \end{bmatrix}. \tag{22}$$

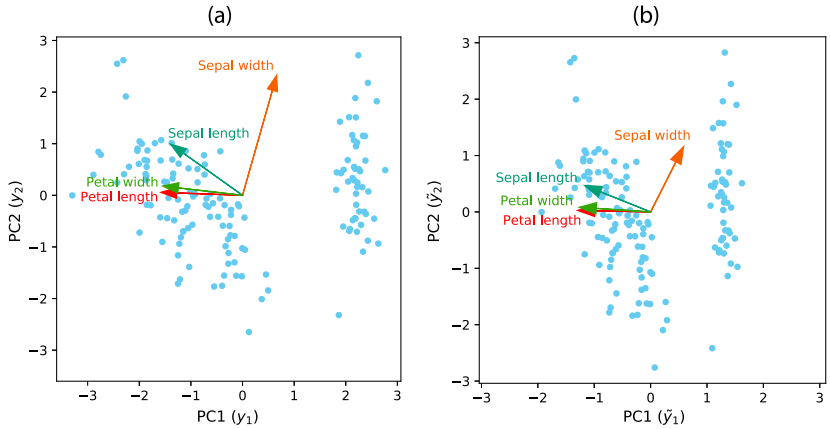

Fig. 8. Visualizing the original variables of the Iris dataset on the respective PCA projection. (a) Vectors representing the projections of the original variables onto the principal components. (b) Biplot containing normalized principal components and loadings.

Two interesting relationships can be inferred from Figure 8(a). First, we have that the angles between the projected features indicate relationships between the original features. For instance, the fact that petal length and petal width yield almost parallel projections indicates that these two features are very similar one another. The second relationship involving the projected data concerns their comparison with the new features $\boldsymbol{y}_1$ (horizontal axis) and $\boldsymbol{y}_2$ (vertical axis). For instance, we have from Figure 8(a) that the petal length vector is inversely oriented to the first principal component, while sepal width is almost parallel to the second principal component. This indicates that petal length contributed greatly to the formation of the first PC, while sepal width contributed more to the second PC.

A closely related way of studying the relationship between original and new variables is based on the concept of a *biplot* [Gabriel 1971]. Figure 8(b) illustrates the biplot obtained for the Iris dataset. There are two main differences between the biplot and the projection shown in Figure 8(a): First, we have that the axes of the biplot are the principal components divided by their respective standard deviation, i.e.,

$$\tilde{\boldsymbol{y}}_1 = \frac{\boldsymbol{y}_1}{\sigma_{\boldsymbol{y}_1}}, \tag{23}$$

$$\tilde{\boldsymbol{y}}_2 = \frac{\boldsymbol{y}_2}{\sigma_{\boldsymbol{y}_2}}. \tag{24}$$

The other difference is that the projections of the original variables are obtained by multiplying a scaled version of the eigenvectors by the original feature's unit vector, that is

$$\left[\text{Sepal length}_{PC_1}\ \text{Sepal length}_{PC_2}\right] = \left[1\ 0\ 0\ 0\right] \begin{bmatrix} \sqrt{\lambda_1} w_{11} & \sqrt{\lambda_2} w_{12} \\ \sqrt{\lambda_1} w_{21} & \sqrt{\lambda_2} w_{22} \\ \sqrt{\lambda_1} w_{31} & \sqrt{\lambda_2} w_{32} \\ \sqrt{\lambda_1} w_{41} & \sqrt{\lambda_2} w_{42} \end{bmatrix}. \tag{25}$$

The motivation for this normalization comes from the fact that the Pearson correlation between $\boldsymbol{y}_j$ and feature $\tilde{\boldsymbol{x}}_k$ is given by

$$PCorr(\boldsymbol{y}_j, \tilde{\boldsymbol{x}}_k) = \sqrt{\lambda_j} w_{kj}. \tag{26}$$

Please refer to Section 7 of the Supplementary Information for a demonstration of this property. Therefore, the projection of each vector shown in Figure 8(b) onto a principal component corresponds to the Pearson correlation coefficient between the respective feature and the principal component in question. As mentioned in Section 3.1, the vector $\boldsymbol{f}_j = (\sqrt{\lambda}_j w_{1j}, \ldots, \sqrt{\lambda}_j w_{4j})^T$ is called the *loadings vector* (or simply loading) of the jth principal component. Furthermore, the angles between the vectors representing the features are reasonable approximations of the correlation between them [Gabriel 1971]. Thus, the biplot is a combination of the score plot, containing the projected values $\tilde{\boldsymbol{y}}_1$ and $\tilde{\boldsymbol{y}}_2$, and the loading plot, containing the loadings of the projection. Therefore, the biplot provides an intuitive visualization of the relationships among the original features and between those and the principal components.

6 OUTLIER DETECTION

The quality of the PCA projection can be quantified with the help of measures aiming at summarizing how well represented the original data is. For instance, if one assumes that the underlying data X follow a multivariate normal distribution, then Hotelling's T-Squared Statistic, T^2 [Hu et al. 2008; Jackson and Mudholkar 1979; Mahmoud et al. 2016], can be used to estimate how likely it is that a particular sample comes from that distribution [Mahmoud et al. 2016]. If we assume that X is centralized (see Section 2), then, after transforming the data through Equation (16), this statistic for the ith object is calculated as

$$T_i^2 = \sum_{j=1}^{\ell} \frac{y_{ij}^2}{\lambda_j}, \tag{27}$$

where ℓ is the number of principal components and $\lambda_1, \lambda_2, \ldots, \lambda_\ell$ are the eigenvalues in decreasing order.

Another approach to measure if a sample is well-adjusted to the projected model involves calculating the *square prediction error* (SPE), also denoted Q [Jackson and Mudholkar 1979; Mahmoud et al. 2016]. To define Q, we use the residue vector R, defined for the ith sample as

$$R_i \equiv \boldsymbol{x}_i' - \boldsymbol{x}_i' \boldsymbol{L}_\ell \boldsymbol{L}_\ell^T, \tag{28}$$

where $\boldsymbol{x}_i'$ is the ith feature vector and $\boldsymbol{L}_\ell$ is a matrix whose columns are the first ℓ loadings vectors (see Section 3). Equation (28) corresponds to attempting to reconstruct the original object $\boldsymbol{x}_i'$ from its projection onto the first ℓ principal components, and then measuring the resulting difference. The total squared residual Q_i associated to the ith object is then calculated as

$$Q_i = R_i R_i^T. \tag{29}$$

To detect outliers, both statistics can be compared with pre-established upper limits (determined by the distributions of T^2 and Q; see, for instance, Jackson and Mudholkar [1979]), and the samples that exceed these values can be considered as exceptions [Mahmoud et al. 2016]. Common visual tools for interpreting these statistics are the T^2 and Q contribution plots, in which values of T_i^2 and Q_i are plotted against the object label number i. In such a plot, outliers are immediately evident as "spikes" in the graph, and one can then set them apart and analyze them more closely.

Expanding upon these ideas, many outlier detection approaches have been proposed. For example, to detect possible anomalies in industrial systems [Hu et al. 2008], for outlier detection in IoT systems [Yu et al. 2017], to identify failures in sensor networks [Chatzigiannakis and Papavassiliou 2007], to detect unusual activities in a smart home [Mahmoud et al. 2012], and many others.

Alternative statistics and models have also been employed to detect outliers via PCA. For instance, singular nodes of a graph can be identified through density functions computed on the

principal components [Costa et al. 2009]. Another alternative is the use of *robust methods*, which consist of fitting a model adjusted by the majority of the samples [Rousseeuw et al. 2006; Rousseeuw and Hubert 2011]. Because the PCA projection can be sensitive to outliers [Rousseeuw and Hubert 2011], some studies replace the covariance matrix by an estimator referred to as *robust covariance* [Hubert et al. 2008; Rousseeuw and Hubert 2011]. Further information regarding robust methods can be found in Hubert et al. [2008].

7 A CASE STUDY

In this section, we show a complete example of a PCA application for data analysis, including the steps of feature extraction, individual and pairwise feature analysis and the interpretation of the principal components and of the respective eigenvalues and eigenvectors. We also discuss this example in the context of data compression.

The objective is to analyze the beans shown in Figure 1. First, we must define a set of features for characterizing the beans. There are many features that could be used for this task; here, we chose seven features, which are calculated for each bean in the picture: the area, perimeter, diameter, elongation, circularity, **average grayscale intensity (Avg. int.)**, and **standard deviation of the grayscale intensity (Std. int.)**. Obtaining these features involves a set of image processing steps for identifying the border of the beans, and also the precise definition for calculating each feature. Since the focus of the example is not the calculation of these features, respective details are not presented here. Please refer to Costa and Cesar Jr [2009] for a more in-depth presentation of the image processing steps. In simple terms, the area, perimeter and diameter features are all related to the size of the beans, but may provide complementary information. For instance, a bean with rough borders will have larger perimeter than another bean with similar area but a smoother border. Regarding the circularity, larger values are obtained for beans having more circular shapes; a maximum value of 1 represents a perfect circle. Elongation is also related to how circular a bean's shape is; however, its values are taken in $[1, +\infty)$, with 1 corresponding to a perfectly round bean. The grayscale intensity is the value of a pixel in an image that was converted to grayscale (sometimes inaccurately called a black-and-white image). Larger values mean that the pixel is "brighter," or closer to the white color. For a dark bean, the average grayscale intensity will be low. The standard deviation of the grayscale intensity is related to the uniformity of the original colors and related intensities of the bean.

The obtained features are organized into a matrix X, each row of which contains the seven values obtained for a specific bean. Each column corresponds to a feature, measured for all beans. Since the image contains 64 beans, X has size 64×7. A table containing the values of matrix X is included in the Supplementary Information. Instead of immediately applying PCA, it is useful to first analyze each feature separately. This can be done by plotting histograms for each of the seven features, as shown in Figure 9.

The histograms show that area and perimeter have similar distributions. The distribution of the elongation has a sharp peak, meaning that most of the beans have very similar elongation. Also, some of the histograms indicate that a few beans have widely distinct features compared to the others. For instance, there is a bean with an elongation much larger than the rest (around 1.7).

It is also important to verify the relationship between the proposed features. For example, if two features are highly correlated, they might be providing redundant information about the beans. The main purpose of PCA is to remove such correlations. The pairwise relationship between features can be analyzed using scatter plots of every combination of two features, as shown in Figure 10.

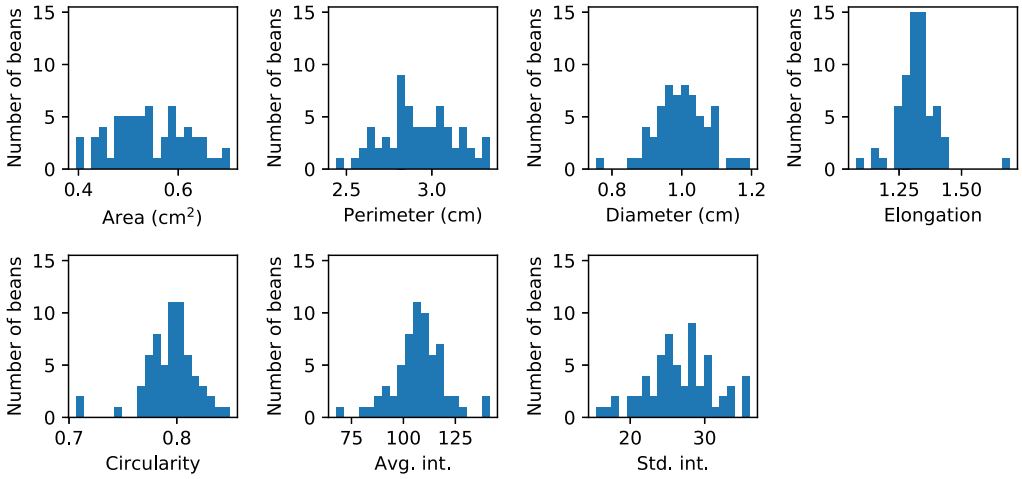

Fig. 9. Histogram of each feature used for characterizing the beans shown in Figure 1.

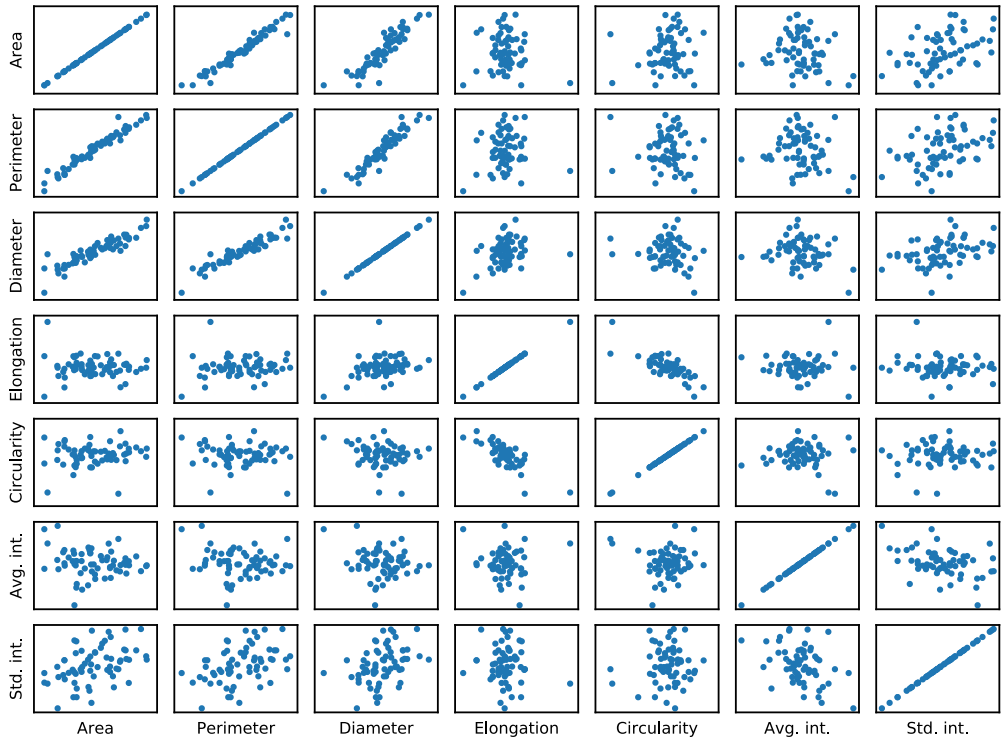

Fig. 10. Scatter plots showing the relationship between pair of features associated with the beans. Each point corresponds to a bean.

The scatter plots show that the area and perimeter are indeed strongly correlated. Both of these features are also correlated with the diameter. The elongation and circularity are also related, since they quantify similar aspects about the shape of the beans. The features associated with the grayscale intensity of the beans are not related to the other five features.

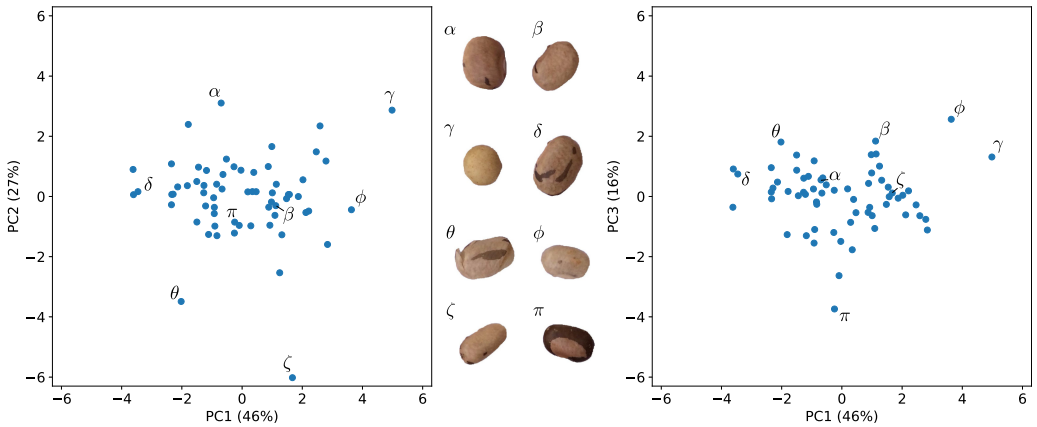

Fig. 11. Principal components of the beans' properties. The left plot shows the first and second components while the right one shows the first and third components. Beans associated to some selected points are shown at the center.

We now proceed to the application PCA itself. Since the features have distinct magnitudes and variances (e.g., area and average grayscale intensity), the features need to be standardized, which is equivalent to computing PCA using the Pearson correlation coefficient matrix. The result is depicted in Figure 11. The two main principal components, PC1 and PC2, are shown on the left plot of Figure 11 and the first and third principal components on the right. Since each point corresponds to a bean, it is possible to visualize the beans associated to a few relevant points. Usually, points having relatively large or small values in the projection are of interest, since they are related to large or small principal component scores and thus can help interpreting the principal components. Some of the beans and their respective positions are shown at the center of Figure 11.

One first important information that can be derived from the projection is the explained variance. Using the concepts presented in Section 3.2, we obtain that the explained variances are 46% for PC1, 27% for PC2, and 16% for PC3, totalling 89% of explained variance for the first three axes, which indicates that the three components provide a good description of the original data. That is because, as observed from the pairwise analysis, some of the features are correlated. If the total explained variance of the three axes were, for instance, 30%, then additional components would be needed in the analysis, and the visualization of the data would be more difficult.

As discussed in Section 5, to interpret the meaning of the principal components, it is useful to analyze the eigenvectors used for projecting the data. The eigenvectors calculated for the beans are shown in Table 2. For the first eigenvector, the area, perimeter and diameter have a much larger weight than the other features. Notice that the sign of the weight (positive or negative) usually does not matter for analyzing the relevance of a feature. However, a negative sign for all three features means that larger values on PC1 correspond to smaller values of area, perimeter and diameter. This can be observed in Figure 11, where beans γ and ϕ, which are small, have larger PC1 values than beans α, δ and θ, which are bigger.

Regarding the second eigenvector, the elongation and circularity have the largest weights. One weight is positive and the other negative, which is related to the fact that a larger elongation means lower circularity. Therefore, PC2 is more closely associated with the elongation of the beans. For instance, beans θ and ζ in Figure 11, which are more elongated, have smaller values of PC2 than beans α and γ.

Table 2. Eigenvectors of the Correlation Matrix Used in the Beans Analysis

Feature	Eigenvector 1	Eigenvector 2	Eigenvector 3
Area	−0.53	0.16	0.23
Perimeter	−0.54	0.01	0.21
Diameter	−0.54	−0.12	0.15
Elongation	−0.10	−0.68	−0.11
Circularity	0.06	0.67	0.09
Avg. int.	0.18	−0.09	0.80
Std. int.	−0.32	0.20	−0.48

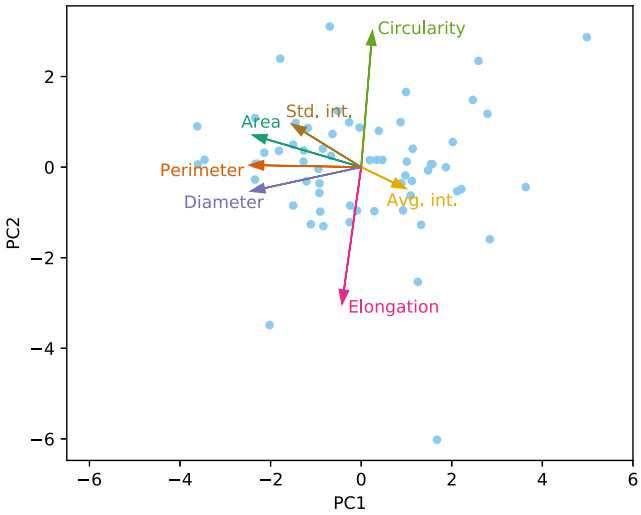

Fig. 12. Biplot of the PCA projection of seven beans features.

Table 2 also shows that the third principal component is mostly related to the average grayscale intensity of the beans, with the standard deviation of intensity also having some contribution. This is why bean π in Figure 11, which is markedly dark, is associated with a very negative value of PC3.

Notice that beans that are very different from the others, such as beans ζ, γ, and π, become outlier points in the principal components. This is one of the reasons why PCA can be used for identifying outlier objects in a dataset. Also, it is important to reiterate that the obtained PCs are uncorrelated. Therefore, the principal component analysis identified three new features, each being associated to a particular aspect of the beans (size, elongation and color). These features are calculated as linear combinations of the original features, with weights given by the eigenvectors shown in Table 2.

Another way of visualizing the projection, presented in Section 5, is the biplot. Figure 12 shows the biplot for the first two principal components. The biplot confirms that the area, perimeter and diameter are negatively associated with PC1, while the circularity and elongation are the main features defining PC2.

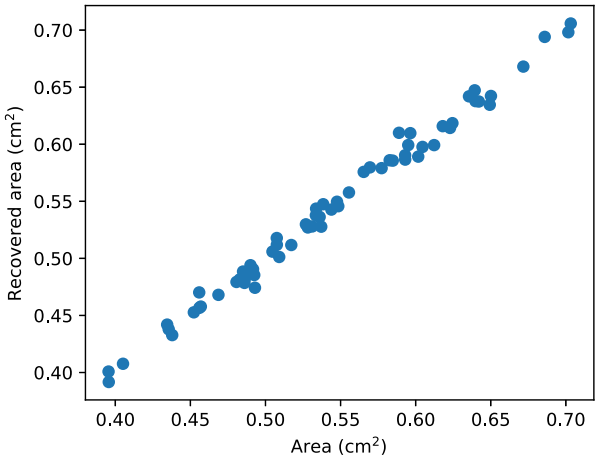

Fig. 13. Relationship between the original areas of the beans and the areas recovered from the first three principal components.

Given that three principal components can be used for representing most of the features calculated for the beans, it is interesting to interpret PCA as a *data compression* method, which is actually one possible application of PCA. Given the principal components, an approximation of the original data can be obtained by calculating

$$X_e = YW^T. \tag{30}$$

Notice that in the equation above, we are using only the first three columns of Y and W. This operation can be seen as the inverse of Equation (9), which was used for calculating the scores. Since the features were normalized, for recovering the scale of the values it is also necessary to add the average of each feature to the respective column of X_e and to multiply the result by the standard deviation of the feature. An example of the relationship between a recovered feature and the respective original feature is shown in Figure 13.

To obtain some quantification of how efficient the compression is, one may proceed as follows. The original data matrix X contains $64 \times 7 = 448$ values. Storing just the first three principal components requires $64 \times 3 = 192$ values, which uses much less storage space than 448. It is also necessary to store the three eigenvectors used in the projection, which amounts to storing $7 \times 3 = 21$ values. In addition, the average and standard deviation of each feature ($7 + 7 = 14$ values) also need to be stored so that the original features can be recovered. Therefore, an approximation of the data can be stored using $192 + 21 + 14 = 227$ values, resulting in nearly a 50% decrease in the required amount of storage space.

8 EXPERIMENTAL STUDY OF PCA APPLIED TO DIVERSE DATASETS

One of the main reasons for the popularity of PCA is its ability to reduce the dimensionality of the original data while preserving as much as possible of its variation. In addition to allowing faster and simpler computational analysis, this reduction also allows the identification of new important variables with distinctive explanatory capabilities. In this section, we apply PCA to some representative real-world datasets from different areas and show the obtained variance explanation as a function of the number of retained principal components. The results provide some indication about typically expected curves obtained when analysing real-world data. Nevertheless, it should

Table 3. Characteristics of the Considered Datasets, Concerning Different Areas

Name	Samples	Features
astronomy (galaxy)	243,500	226
astronomy (ionosphere)	351	34
biology (gene)	545	79
biology (leaf)	340	15
chemistry (milk)	86	8
chemistry (wine)	178	13
computer (machine)	209	8
computer (segment-challenge)	1500	19
engineering (energy)	2,049,280	7
engineering (slump)	103	10
geography (dengue)	1,986	12
geography (forest)	517	11
linguistics (reviews)	1,500	10,000
linguistics (blog)	52,397	281
materials (glass)	214	9
materials (plates)	1,941	27
medicine (diabetes)	768	8
medicine (parkinsons)	195	22
meteorology (el niño)	533	7
meteorology (ozone)	1,847	72

be noted that the specific shape of the curve depends on the dataset and, in particular, on the features used in the analysis. We considered two datasets representative of each of 10 different fields. The selected areas are astronomy, biology, chemistry, computer science, engineering, geography, linguistics, materials science, medicine, and meteorology. After the required pre-processing for eliminating incomplete and categorical data, PCA was applied to the datasets.

8.1 Dataset Selection

For all considered datasets, we eliminated non-numerical data, such as categorical values and dates. The features were also standardized to have zero mean and unit variance. Some characteristics of the considered datasets are shown in Table 3.

In astronomy, we considered data from *galaxies* [Willett et al. 2013]; more specifically, a table that comprises features of spectroscopic redshifts for the galaxies. The second dataset comprised the *ionosphere* as measured by radar [Sigillito et al. 1989]. In biology, we considered datasets regarding *gene* expression levels [Eisen et al. 1998] and morphological features of *leaves* [Dheeru and Karra Taniskidou 2017; Silva et al. 2013]. Two datasets of food were employed representing chemistry: (i) data on characteristics of *wine* [Dheeru and Karra Taniskidou 2017; Forma et al. 1988] and (ii) data on *milk* composition [Daudin et al. 1988]. For computer science, two different subjects were considered, namely, hardware characteristics of computers [Dheeru and Karra Taniskidou 2017] (*machine*) and features of image *segmentation* data. This image segmentation (*segment-challenge*) dataset is part of the datasets provided by the software Weka [Witten et al. 2016]. In engineering, we used information on electric power consumption by households [Dheeru and Karra Taniskidou 2017] (*energy*) and a set of concrete *slump* tests [Dheeru and Karra Taniskidou 2017; Yeh 2006].

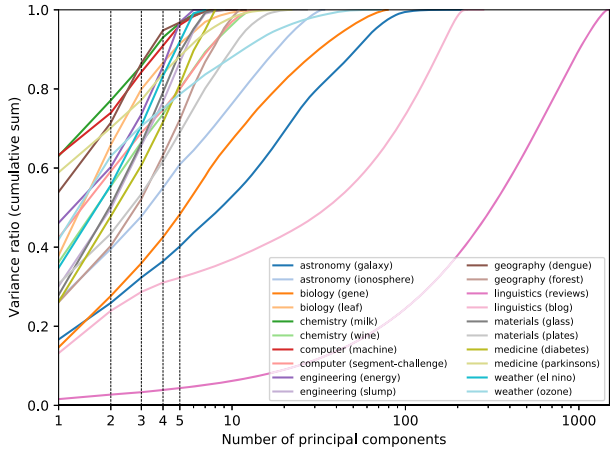

Fig. 14. Cumulative explained variance ratio according to the number of principal components retained, for all the datasets presented in Table 3. Vertical dashed lines indicate the situations where the first 2, 3, 4, and 5 PCA components are retained.

The geography datasets contain data on spatial coordinates and information related to weather. The first one is about *dengue* fever [Hales et al. 2002], and the second concerns *forest* fires [Cortez and Morais 2007; Dheeru and Karra Taniskidou 2017]. In linguistics, the first dataset comprises the frequency of linguistic elements (e.g., punctuation and symbols) in texts of commerce *reviews* [Dheeru and Karra Taniskidou 2017; Liu et al. 2011]; the second one contains statistics of *blog* feedbacks [Buza 2014; Dheeru and Karra Taniskidou 2017]. The dataset considered in materials science is about *glass* identification [Evett and Ernest 1987], with information on refractive index and chemical composition, and features regarding *plates* faults [Dheeru and Karra Taniskidou 2017; Semeion 2018]. In medicine, the selected dataset is about the clinical characteristics of people with or without *diabetes* [Smith et al. 1988], and the other dataset considers voice features of patients with or without *Parkinson*'s disease [Dheeru and Karra Taniskidou 2017; Little et al. 2007]. Finally, the meteorology datasets are: (i) environmental features of *El Niño* [Bay et al. 2000; Dheeru and Karra Taniskidou 2017] and (ii) features related to *ozone* levels in the atmosphere [Dheeru and Karra Taniskidou 2017].

8.2 Results and Discussion

To compare the amount of variance retained by PCA for the different datasets, Figure 14 shows the number of PCA components against the respective cumulative explained variance ratio, defined by Equation (20). In the majority of cases, the first three principal components represent more than 50% of the variance in the datasets.

As a complementary analysis, we also show the cumulative explained variance ratio among the datasets by normalizing the number of principal components retained by the number of dimensions in the original dataset (see Figure 15). The main purpose of this analysis is to identify the variance retained as a function of the reduction in the dimensionality of the dataset. By comparing Figures 15 and 14, we note that the relative order of the curves change. This means that some datasets need many PCA components to achieve large variance ratio, but since these datasets originally have a large number of features, only a small percentage of the number of features is enough to represent most of the variance in the data. Note that the variance explanation is lower

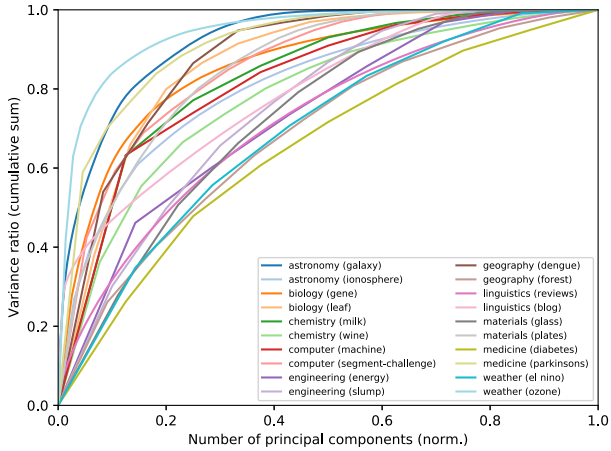

Fig. 15. Cumulative explained variance ratio according to the normalized number of principal components retained for all the datasets presented in Table 3. Note that the normalized number of principal components consists of dividing the number of principal components by the total number of features in the original dataset.

Table 4. Main Typical Applications of the Related Methods Described Here

Method	Main application	Main references
2DPCA	For data that are naturally described as matrices, such as images	[Yang et al. 2004]
Sparse PCA	When there are more features than objects in the data	[Zou et al. 2006]
Incremental PCA	Allow a fast update of PCA when new data is added to the dataset	[Cardot and Degras 2018]
Kernel PCA	When the feature space comprises nonlinear features	[Schölkopf et al. 1997]

for curves near the diagonal. The dataset having the nearest curve to the diagonal is *medicine (diabetes)*.

9 RELATED METHODS

In this section, we provide an overview of some alternative approaches for dimensionality reduction that bear some resemblance to PCA. Table 4 summarizes the described approaches, as well as their main applications.

9.1 Sparse PCA

Though often used as a dimensionality reduction method, PCA itself is not immune to the *curse of dimensionality*: the geometry and statistics of high-dimensional spaces are radically counter-intuitive to our three-dimensional minds—see, for instance, Chapter 1 in Giraud [2015]. In the context of PCA, the *curse of dimensionality* manifests itself as two separate issues. First, the small but non-vanishing loadings of the principal components become increasingly difficult to interpret: what constitutes a "negligible" contribution given tens of thousands of features, as in some types of data? Second, as the number of features becomes comparable to, or larger than, the number of

objects, our ability to correctly estimate the covariance matrix from data decreases dramatically—to the point that estimated principal components are orthogonal to the *actual* principal components [Johnstone and Lu 2009; Nadler 2008; Paul 2007], and estimated eigenvalues follow a distribution (known as the Marchenko-Pastur Law; see Chapter 2 in Anderson et al. [2009] and Vershynin [2012]) that is only loosely connected to the underlying covariance structure.

To overcome these issues, **sparse PCA (SPCA)** was developed to ensure that less significant loadings be *exactly* zero [Zou et al. 2006]. By recasting PCA as a least-squares regression problem (see, for instance, Section 3.4 in Hastie et al. [2009]), one can make use of standard regularization techniques to enforce zero coefficients in the regression—which correspond to zero loadings of the sparse principal components. More specifically, suppose one is given the centralized (i.e., with every feature having mean zero) data matrix X with its objects $x_1', \ldots, x_p'$. Then, to obtain the first k sparse principal components, one must solve the optimization problem,

$$(A^*, B^*) = \arg\min_{A,B} \sum_{i=1}^{p} \|\boldsymbol{x}_i' - \boldsymbol{x}_i' A B^T\|^2 + \gamma \sum_{j=1}^{k} \|\boldsymbol{\beta}_j\|^2 + \sum_{j=1}^{k} \gamma_{1,j} \|\boldsymbol{\beta}_j\|_1, \tag{31}$$

subject to $A^T A = I$, where A is an $n \times k$ matrix. Here, $\boldsymbol{\beta}_j$ are the column vectors of B, which is an $n \times k$ matrix whose columns correspond to the sparse principal components, and $\|\cdot\|_1$ denotes a vector's "Manhattan norm" $\|\boldsymbol{x}\|_1 = |\boldsymbol{x}_1| + |\boldsymbol{x}_2| + \cdots + |\boldsymbol{x}_n|$. The free parameters γ and $\gamma_{1,1}, \ldots, \gamma_{1,k}$ correspond to *ridge* and *lasso* regularization penalties, respectively, and are always non-negative.

An interested reader can consult Zou and Hastie [2005]; Zou et al. [2006] for the computational intricacies of solving Equation (31). That being said, implementations of SPCA can already be found in the `elasticnet` (for R) and `sklearn` (for Python) packages.

9.2 2DPCA

All the PCA calculations considered so far represent each object as a feature vector of dimension n. After the projection, the object can then be represented as a new vector of dimension ℓ. The use of feature vectors inherently constrains the representation of the object to be one-dimensional. For instance, if the objects are images having r rows and c columns, each image needs to be transformed into 1D vectors of size $r \times c$ by concatenating the rows or columns of the image [Turk and Pentland 1991]. This procedure removes spatial relationships among the pixels in the image, which could have been used for improving the quality of the projection. Thus, alternative PCA approaches have been defined for objects that are naturally described as matrices or tensors [Lu et al. 2008; Ye et al. 2004]. Such methods can lead to better compression and data reconstruction as well as to lower computer memory requirements.

A common approach for projecting matrices instead of vectors, known as 2DPCA [Yang et al. 2004], works as follows. Recall that the first principal component is such that the variance of the projected points is maximum. Also, each 1D vector, representing a respective object, becomes a scalar (a number) after the projection onto the first principal component. If instead of a 1D vector the object is represented as a $r \times c$ matrix A_i, then it is possible to define a vector $\boldsymbol{w}_1$ having size c that projects this image into a new vector $\boldsymbol{y}_i'$ with size r, that is,

$$(\boldsymbol{y}_i')^T = A_i \boldsymbol{w}_1. \tag{32}$$

Therefore, as in PCA, one can search for the vector $\boldsymbol{w}_1$ that leads to the maximum variance of the projected vectors. Since the variance is calculated over a set of numbers and each $\boldsymbol{y}_i'$ is a

vector, it is necessary to define a slightly different approach for measuring the variance of the projected data, based on the covariance matrix of the $\boldsymbol{y}_i'$ vectors. Nevertheless, it can be shown [Yang et al. 2004] that the best choice of $\boldsymbol{w}_1$ is the eigenvector associated with the largest eigenvalue of matrix

$$K_t = \frac{1}{p}\sum_{i=1}^{p}(A_i - \bar{A})^T(A_i - \bar{A}), \tag{33}$$

where $\bar{A}$ is a matrix containing the element-wise average of all input matrices, that is,

$$\bar{A} = \frac{1}{p}\sum_{i=1}^{p} A_i. \tag{34}$$

Similar to the eigenvectors of the covariance matrix in the PCA method, the eigenvectors of K_t associated to the ℓ-largest eigenvalues lead to the optimal projection for the 2DPCA.

9.3 Incremental PCA

PCA is usually employed in dimensionality reduction of a given dataset so that the reduced data can be more easily analyzed or fed to another algorithm. The original method has memory usage of at least $O(pn)$, as it is necessary to hold all the data in memory, where p is the sample size and n the number of features. If PCA is obtained through eigenvalue decomposition, then additional space of $O(n^2)$ is required in memory for storing the covariance matrix. The traditional PCA algorithm requires all the data to be available to obtain the principal components. Thus, it is often considered a *batch* or *offline* method. The batch PCA can be performed through eigenvalue decomposition or **singular value decomposition (SVD)**, both can be computed in $O(pn\min(p,n))$ floating point operations [Arora et al. 2012; Cardot and Degras 2018].

For particularly massive datasets, with thousands or millions of objects, p, and features n, memory usage can be a problem. The implied computation costs also limits real-time applications. One example of this scenario is visual learning and recognition that uses the compressed PCA output as an input for machine learning algorithms [Artac et al. 2002].

To solve these problems, ***Incremental PCA (IPCA)*** methods were proposed [Haitao Zhao et al. 2006; Juyang Weng et al. 2003]. These methods have been created to allow the calculation of the principal components in an incremental way. Furthermore, IPCA methods enable the principal components to be updated when new data is added to the dataset. However, the results of these algorithms might not be entirely accurate, since the results obtained from IPCA and batch PCA may be distinct. IPCA methods differ in terms of statistical accuracy for different types of datasets and computational cost. An interesting comparative analysis of IPCA can be found in Cardot and Degras [2018].

As an example of an IPCA method, we present an approach suggested by Arora et al. [2012], which is based on the incremental SVD of Brand [2002]. This method allows eigenvalues to be obtained from the covariance matrix $Cov(X)$ after adding a new data vector using only the previously calculated eigenvalues. Thus, the method does not require calculating all the eigenvalues from the updated data matrix, which can significantly improve the computation speed.

This approach starts with $\Delta_p = \tilde{W}_p\tilde{\Lambda}_p\tilde{W}_p^T$, an approximation of the covariance matrix K_p relative to p data points, where $\tilde{W}_p$ is an approximation of the $n \times \ell$ reduced transformation matrix, W_p, and $\tilde{\Lambda}_p$ is a diagonal matrix with the approximated first ℓ eigenvalues of K_p. When a new data point

x'_{p+1} appears, the matrix Δ_p can be updated by decomposing the centralized vector $\hat{\boldsymbol{x}}'_{p+1} = \boldsymbol{x}'_{p+1} - \mu_X$ as $\hat{\boldsymbol{x}}'_{p+1} = \boldsymbol{c}_{p+1}\tilde{W}_p + (\hat{\boldsymbol{x}}'_{p+1})^{\perp}$, where $\boldsymbol{c}_{p+1} = \hat{\boldsymbol{x}}'_{p+1}\tilde{W}_p^T$ are the coordinates of $\hat{\boldsymbol{x}}'_{p+1}$ projected in the ℓ-dimensional space spanned by $\tilde{W}_p$, and $(\hat{\boldsymbol{x}}'_{p+1})^{\perp}$ is the projection onto the space orthogonal to $\tilde{W}_p$. A new approximation of the covariance matrix, Δ_{p+1}, is given by

$$\Delta_{p+1} = \frac{p}{p+1}\Delta_p + \frac{p}{(p+1)^2}(\hat{\boldsymbol{x}}'_{p+1})^T\hat{\boldsymbol{x}}'_{p+1}. \tag{35}$$

We can rewrite this equation as

$$\Delta_{p+1} = \left[\frac{(\hat{\boldsymbol{x}}'_{p+1})^{\perp}}{\| (\hat{\boldsymbol{x}}'_{p+1})^{\perp} \|}\tilde{W}_p\right] H_{p+1} \left[\frac{(\hat{\boldsymbol{x}}'_{p+1})^{\perp}}{\| (\hat{\boldsymbol{x}}'_{p+1})^{\perp} \|}\tilde{W}_p\right]^T, \tag{36}$$

with the $\ell + 1 \times \ell + 1$ matrix H_{p+1} given by

$$H_{p+1} = \frac{p}{(p+1)^2}\begin{bmatrix} (p+1)\tilde{\Lambda}_p + c_{p+1}^T\boldsymbol{c}_{p+1} & \| (\hat{\boldsymbol{x}}'_{p+1})^{\perp} \| \, \boldsymbol{c}_{p+1}^T \\ \| (\hat{\boldsymbol{x}}'_{p+1})^{\perp} \| \, \boldsymbol{c}_{p+1} & \| (\hat{\boldsymbol{x}}'_{p+1})^{\perp} \|^2 \end{bmatrix}. \tag{37}$$

By performing the eigendecomposition of H_{p+1} and discarding the component related to the smallest eigenvalue, it is possible to obtain the ℓ eigenvectors of Δ_{p+1}, and with that the new approximated reduced transformation matrix $\tilde{W}_{p+1}$. As can be seen, the new complete data matrix was not used in this method. It is important to note that the method only performs updates concerning a single data vector; in Levy and Lindenbaum [1998], a similar approach for block updates can be found.

9.4 Kernel PCA

PCA consists of a linear projection that takes into account the optimization of the data covariance matrix. However, many datasets cannot be represented using a linear projection [Andreas 1998]. To deal with nonlinear features, *kernel* PCA was developed [Schölkopf et al. 1997]. This projection consists of an extension of the original method [Schölkopf et al. 1997] and has been employed mainly in pattern recognition problems. Its method aims at transforming a dataset that cannot be linearly separated into a higher-dimensional, linearly separable dataset.

First, a nonlinear mapping ϕ is chosen to map our objects into a higher-dimensional (sometimes even infinite-dimensional) vector space, and a kernel function κ is defined as the scalar product in this new space. The kernel matrix, K, can be calculated as

$$K_{ij} = \kappa(\boldsymbol{x}'_i, \boldsymbol{x}'_j) = \phi(\boldsymbol{x}'_i)^T\phi(\boldsymbol{x}'_j), \; i, j = 1, \ldots p, \tag{38}$$

where $\boldsymbol{x}'_i$ and $\boldsymbol{x}'_j$ are, respectively, the samples i and j of the data matrix X. In contrast with the covariance matrix, in kernel PCA the dimensions of K are $p \times p$, in which p is the number of samples represented in X. In the following, K shall be centered at zero; this can be accomplished, in matrix form, as

$$\tilde{K} = K - \mathbf{1}_p K - K\mathbf{1}_p + \mathbf{1}_p K \mathbf{1}_p, \tag{39}$$

where the matrix $\mathbf{1}_p$ is a $p \times p$ square matrix with all elements being $1/p$ [Bishop 2006]. Finally, the projection is calculated by solving the following eigenproblem:

$$\tilde{K}\boldsymbol{\alpha}_k = \lambda_k \boldsymbol{\alpha}_k, \tag{40}$$

where $\boldsymbol{\alpha}_k$ is a p-dimensional column vector and λ_k is a scalar. Then, each sample $\boldsymbol{x}'_i$ can be projected as

$$y_{ij} = \sum_{k=1}^{p} \alpha_{kj} \kappa(\boldsymbol{x}'_k, \boldsymbol{x}'_i), \; j = 1, \ldots, \ell, \tag{41}$$

where α_{kj} is the kth entry of the vector $\boldsymbol{\alpha}_j$ and $\ell \leq n$ is the number of retained dimensions.

Notice that, in all the calculations above, the nonlinear mapping ϕ appears only in scalar products. Thus, the main idea in kernel PCA is to bypass any calculations that require its explicit use, and instead to compute the products via the kernel function κ, for which an explicit form is actually given. Different choices of kernel function correspond to different scalar products in different higher-dimensional spaces, and each choice lends itself to different data distributions. A wide array of options has been employed in kernel PCA, such as Gaussian [Mika et al. 1999], polynomial [Kim et al. 2002], sigmoid [Mansouri et al. 2016], and others [Cheng et al. 2010; Liu 2004; Schölkopf et al. 2002].

10 OTHER APPROACHES

In this section, we briefly outline other popular methods for some form of dimensionality reduction. Since each has its own advantages and drawbacks, a straightforward comparison between them is difficult. For instance, PCA would most likely outperform all of these methods with respect to variance preservation (by its very definition). Unlike PCA, all the methods below are nonlinear.

10.1 t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a dimensionality reduction method originally presented in van der Maaten and Hinton [2008]. Its main objective is data visualization, and it approaches the problem by preserving (as best as possible) the *distribution* of distances from a point to its neighbors.

To accomplish this, one considers the joint probability p_{ij} for the distance between the feature vectors of objects i and j. We wish to approximate the distribution of the p_{ij} by the distribution of the values q_{ij} of the distances between the projected feature vectors $\boldsymbol{y}'_i$. However, while the p_{ij} are modelled using a Gaussian distribution, the q_{ij} employ a Student t distribution with a single degree of freedom (hence, the name *t-Distributed* SNE).

The next step is to set up the projected feature vectors $\boldsymbol{y}'_i$ so that the distribution of q_{ij} approximates that of p_{ij}. In other words, one aims at minimizing the Kullback-Leibler divergence between these distributions, which can be performed by standard methods such as gradient descent; see [van der Maaten and Hinton 2008, Subsection 3.4].

Unlike the principal components in PCA, the new axes in a t-SNE projection are not functions of the original features, and thus have no intrinsic meaning in an of themselves. However, due to its proposal of "preserving the distribution of distances between points," t-SNE is often a better choice than PCA when the objective is to identify groups and clusters in the data.

10.2 UMAP

In terms of dimensionality reduction for data visualization, **Uniform Manifold Approximation and Projection (UMAP)** could be called the current state of the art. It was first developed in McInnes et al. [2018] and has quickly established itself as a popular data visualization method.

Its underlying assumption is that the data are not randomly scattered on a high-dimensional feature space, but instead lie (up to small perturbations) on a low-dimensional embedded manifold (one can think of manifolds as generalized surfaces). The goal, then, is to estimate this manifold's

underlying structure, and then to set up the projection so that it approximates this structure as best as possible; for UMAP's intents and purposes, the manifold's "structure" is represented by its Riemannian metric and by its simplicial structure. The precise details of how to do this are beyond the scope of this survey, and so we refer the reader to McInnes et al. [2018].

Despite coming from a different motivation, UMAP often accomplishes a similar goal to t-SNE of "preserving distributions of distances." In fact, from an algorithmic point of view, these two methods have more in common than any others on this section (see Section 3 and Appendix C in McInnes et al. [2018]).

10.3 Autoencoders

Autoencoders are a particular kind of neural network in which, after passing through one or more hidden variable layers of lower dimension, the output activation is required to be as close as possible to the input data [Tomar et al. 2017; Vincent et al. 2008]. The name for this method derives from its motivation in coding theory: given an input message, one aims to send it through some channel of limited capacity, requiring it to be encoded into a smaller message (usually called the *code*). At the end, the code must be decoded, and the recovered message should be as close as possible to the original one.

In more concrete terms, given our input data $\boldsymbol{x}'_1, \boldsymbol{x}'_2, \ldots, \boldsymbol{x}'_p$, which each $\boldsymbol{x}'_j$ having dimension n, we first map them into the "hidden" variables $\boldsymbol{y}'_1, \boldsymbol{y}'_2, \ldots, \boldsymbol{y}'_p$, where each $\boldsymbol{y}'_j$ has dimension $\ell < n$. Then, taking these new objects, we attempt to reconstruct our original data, obtaining the "output" variables $\boldsymbol{z}'_1, \boldsymbol{z}'_2, \ldots, \boldsymbol{z}'_p$ again of dimension n.

Since this is a neural network, each mapping from $\boldsymbol{x}'$ to $\boldsymbol{y}'$ and back to $\boldsymbol{z}'$ has its own parameters to be fine-tuned in an optimization process. To carry this out, one has to minimize the difference between output and input variables, usually measured by a standard loss function such as the average distance between them. Interestingly, although we use the standard neural network nomenclature of "hidden" and "output" variables, when using autoencoders for dimensionality reduction, we are actually interested in the variables $\boldsymbol{y}'_j$, which are effectively our sought-after, lower-dimensional representation of the data.

11 CONCLUDING REMARKS

PCA has become a standard approach in data analysis as a consequence of its ability to reduce dimensionality, through decorrelation, while preserving data variance. In this work, we reported an integrated and systematic review of PCA covering several of its theoretical and applied aspects. We started by providing a simple application example of PCA to real-world data (beans). Next, we developed the concept of PCA from more basic aspects of multivariate statistics, and presented the important issues of variance preservation. The option to normalize or not the original data was addressed subsequently, and it was shown that each of these alternatives can have a major impact on the obtained results. Then, we discussed how to interpret the eigenvectors calculated during PCA through loadings and biplots. Afterwards, we briefly explained how to use PCA for outlier detection in datasets. We also illustrated the potential of PCA for variance explanation and dimensionality reduction for several real-world datasets, including a complete example of data analysis with PCA. Our analysis of diverse real-world data indicated, at least for the considered data, a wide variety of variance concentration patterns along the first principal components across different fields. Last but not least, we provided an introductory outlook to several complementary alternatives to PCA.

All in all, we hope that the reported work on PCA, covering from basic principles to experimental investigations of variance explanation and alternative approaches, has motivated the reader to probe further regarding the diverse theoretical and applied aspects of PCA.

APPENDIX

A LIST OF SYMBOLS

Table 5. Description of the Main Symbols Used in This Work

Definition	Symbol	Type of data
Number of features	n	Scalar
Number of retained principal components	ℓ	Scalar
Number of objects	p	Scalar
Data matrix	X	Matrix
jth feature for all objects	$\boldsymbol{x}_j$	Column vector
Feature vector of object i	$\boldsymbol{x}'_i$	Row vector
jth feature of object i in matrix X	x_{ij}	Scalar
Average of feature $\boldsymbol{x}_j$	$\mu_{\boldsymbol{x}_j}$	Scalar
Standard deviation of feature $\boldsymbol{x}_j$	$\sigma_{\boldsymbol{x}_j}$	Scalar
Transformed data matrix	Y	Matrix
jth transformed feature (principal component) for all objects	$\boldsymbol{y}_j$	Column vector
Transformed feature vector of object i	$\boldsymbol{y}'_i$	Row vector
PCA transformation matrix	W	Matrix
jth column of W	$\boldsymbol{w}_j$	Column vector
Covariance between features $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$	$\text{Cov}(\boldsymbol{x}_i, \boldsymbol{x}_j)$	Scalar
Pearson correlation between features $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$	$\text{PCorr}(\boldsymbol{x}_i, \boldsymbol{x}_j)$	Scalar
Covariance of data matrix X	$\text{Cov}(X)$	Matrix
Explained variance of principal component i	EV_i	Scalar
Cumulative explained variance of the retained principal components	CV	Scalar
Percentage of CV compared to the total variance of the data	G	Scalar

B BASIC MATHEMATICAL CONCEPTS

Vectors and matrices, important mathematical concepts, are typically treated more formally as part of Linear Algebra. In particular, vectors are understood as elements of *vector spaces* (also called *linear spaces*), which is a mathematical formalization aimed at endowing a set of elements with relatively intuitive properties. More informally speaking, vectors in the vector space $\mathbb{R}^n$ are tuples $\boldsymbol{x}$ containing n entries, each entry being a real number. The closure property of a vector space, for instance, ensures that the result of additions between two vectors, or the product of any vector by a scalar α (by scalar, we mean here any real number), are also vectors that belong to that same space. The reader is referred to (e.g., Strang [1993]), for a more complete description of vector spaces and their properties.

Another important concept related to vectors is that of a basis. Intuitively, a *basis* of a vector space is a set of its vectors so that any other vector in this space can be obtained as a *linear combination* (weighted sum) of the vectors in the basis. A linear combination of vectors is any expression of the form

$$\alpha_1 \boldsymbol{x}_1 + \alpha_2 \boldsymbol{x}_2 + \cdots + \alpha_m \boldsymbol{x}_m,$$

where m is a non-negative integer, each α_i is a scalar and each $\boldsymbol{x}_i$ is vector of our vector space. Notice that the linear combination yields a new vector in the same space.

Two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are said to be *orthogonal* if their inner product is null. A set of vectors is said to be orthogonal if all their pairwise combinations are orthogonal. In particular, the vectors composing a basis of a vector space can be orthogonal.

A particularly important operation in a vector space is the *linear transformation* of a given row vector $\boldsymbol{x}$ (written as a $1 \times n$ matrix), which can be generally expressed as the action of an $n \times n$ square matrix on that vector, i.e.,

$$\boldsymbol{y} = \boldsymbol{x}A. \tag{42}$$

Observe that each element i of $\boldsymbol{y}$ corresponds to a linear combination of the elements of $\boldsymbol{x}$, with weights given by the ith column of A. Since vectors x and y have size $1 \times n$, they are called *row vectors*.

Given a linear transformation defined by a respective matrix A, it is possible to consider the respective eigenvalues, λ_j, and eigenvectors, $\boldsymbol{w}_j$, (e.g., da F Costa [2020]), so that

$$A\boldsymbol{w}_j = \lambda_j \boldsymbol{w}_j. \tag{43}$$

Thus, an eigenvector of a matrix A is a special vector, in that it does not change its direction (though it typically changes the magnitude by λ_j) under action by A. An $n \times n$ matrix A can have up to n distinct eigenvalues, with respective eigenvectors, but it often does not. Also, vector $\boldsymbol{w}_j$ is a *column vector*, since it has size $n \times 1$.

If case A is real and symmetric, then all its (not necessarily distinct) eigenvalues will be real, and the respective eigenvectors will be mutually orthogonal. An example of this type of matrix is the covariance matrix, which is real and symmetric.

REFERENCES

Hervé Abdi and Lynne J. Williams. 2010. Principal component analysis. *Wiley Interdisc. Rev.: Comput. Stat.* 2, 4 (2010), 433–459.

Greg W. Anderson, Alice Guionnet, and Ofer Zeitouni. 2009. *An Introduction to Random Matrices*. Cambridge University Press, Cambridge, UK.

König Andreas. 1998. A survey of methods for multivariate data projection, visualisation and interactive analysis. In *Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems*. Citeseer, 55–59.

Raman Arora, Andrew Cotter, Karen Livescu, and Nathan Srebro. 2012. Stochastic optimization for PCA and PLS. In *Proceedings of the 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton'12)*. 861–868. DOI: https://doi.org/10.1109/Allerton.2012.6483308

M. Artac, M. Jogan, and A. Leonardis. 2002. Incremental PCA for on-line visual learning and recognition. In *Object Recognition Supported by User Interaction for Service Robots*, Vol. 3. IEEE Comput. Soc, Quebec City, Que., Canada, 781–784. DOI: https://doi.org/10.1109/ICPR.2002.1048133

Stephen D. Bay, Dennis Kibler, Michael J. Pazzani, and Padhraic Smyth. 2000. The UCI KDD archive of large data sets for data mining research and experimentation. *ACM SIGKDD Explor. Newslett.* 2, 2 (2000), 81–85.

Gordon Bell, Tony Hey, and Alex Szalay. 2009. Beyond the data deluge. *Science* 323, 5919 (2009), 1297–1298.

Dimitri P. Bertsekas and John N. Tsitsiklis. 2002. *Introduction to Probability*. Vol. 1. Athena Scientific, Belmont, MA.

Christopher M. Bishop. 1999. Bayesian PCA. In *Advances in Neural Information Processing Systems*. MIT Press, 382–388.

Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning*. Springer.

Matthew Brand. 2002. Incremental singular value decomposition of uncertain data with missing values. In *Proceedings of the European Conference on Computer Vision (ECCV'02)*, Gerhard Goos, Juris Hartmanis, Jan van Leeuwen, Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen (Eds.). Vol. 2350. Springer, Berlin, 707–720. DOI: https://doi.org/10.1007/3-540-47969-4_47

Richard G. Brereton. 2018. *Principal Component Analysis and Unsupervised Pattern Recognition*. John Wiley & Sons, Chapter 4, 163–214. DOI: https://doi.org/10.1002/9781118904695.ch4 Retrieved from arXiv: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781118904695.ch4.

Rasmus Bro and Age K. Smilde. 2014. Principal component analysis. *Anal. Methods* 6, 9 (2014), 2812–2831.

Krisztian Buza. 2014. Feedback prediction for blogs. In *Data Analysis, Machine Learning and Knowledge Discovery*. Springer, 145–152.

Hervé Cardot and David Degras. 2018. Online principal component analysis in high dimension: Which algorithm to choose?: Online PCA in high dimension. *Int. Stat. Rev.* 86, 1 (Apr. 2018), 29–50. DOI: https://doi.org/10.1111/insr.12220

Vassilis Chatzigiannakis and Symeon Papavassiliou. 2007. Diagnosing anomalies and identifying faulty nodes in sensor networks. *IEEE Sensors J.* 7, 5 (2007), 637–645.

Chun-Yuan Cheng, Chun-Chin Hsu, and Mu-Chen Chen. 2010. Adaptive kernel principal component analysis (KPCA) for monitoring small disturbances of nonlinear processes. *Industr. Eng. Chem. Res.* 49, 5 (2010), 2254–2262.

Paulo Cortez and Aníbal de Jesus Raimundo Morais. 2007. A data mining approach to predict forest fires using meteorological data. In *Proceedings of the 13th Portuguese Conference on Artificial Intelligence (EPIA'07)*, 512–523.

Luciano da Fontoura Costa and Roberto Marcondes Cesar Jr. 2009. *Shape Classification and Analysis: Theory and Practice*. CRC Press, Boca Raton.

L da F. Costa, Francisco A. Rodrigues, Claus C. Hilgetag, and Marcus Kaiser. 2009. Beyond the average: Detecting global singular nodes from local features in complex networks. *Europhys. Lett.* 87, 1 (2009), 18008.

Pádraig Cunningham. 2008. Dimension reduction. In *Machine Learning Techniques for Multimedia*. Springer, 91–112.

Luciano da F. Costa. 2020. Eigenvalues, Eigenvectors (CDT-28). Retrieved from https://www.researchgate.net/publication/340628834_Eigenvalues_Eigenvectors_CDT-28.

J. J. Daudin, C. Duby, and P. Trecourt. 1988. Stability of principal component analysis studied by the bootstrap method. *Statistics: J. Theor. Appl. Stat.* 19, 2 (1988), 241–258.

Dua Dheeru and Efi Karra Taniskidou. 2017. UCI Machine Learning Repository. Retrieved from http://archive.ics.uci.edu/ml.

Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein. 1998. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. U.S.A.* 95, 25 (1998), 14863–14868.

Kim Esbensen and P. Geladi. 2009. *Principal Component Analysis: Concept, Geometrical Interpretation, Mathematical Background, Algorithms, History, Practice*. Vol. 2. 211–226. DOI: https://doi.org/10.1016/B978-044452701-1.00043-0

Brian Everitt and Anders Skrondal. 2002. *The Cambridge Dictionary of Statistics*. Vol. 106. Cambridge University Press, Cambridge.

Ian W. Evett and J. Spiehler Ernest. 1987. Rule induction in forensic science. Central research establishment. Home office forensic science service. Aldermaston. *Reading, Berkshire RG7 4PN* (1987).

Willliam Feller. 2008. *An Introduction to Probability Theory and Its Applications*. Vol. 2. John Wiley & Sons.

Kitty Ferguson. 2002. *Tycho and Kepler: The Unlikely Partnership that Forever Changed our Understanding of the Heavens*. Bloomsbury Publishing, New York, NY.

Ronald A. Fisher. 1936. The use of multiple measurements in taxonomic problems. *Ann. Hum. Genet.* 7, 2 (1936), 179–188.

Imola K. Fodor. 2002. *A Survey of Dimension Reduction Techniques*. Technical Report. Lawrence Livermore National Laboratory, Berkeley, CA.

M. Forma, R. Leardi, C. Armanino, S. Lanteri, P. Conti, and P. Princi. 1988. *PARVUS, an Extendable Package of Programs for Data Exploration, Classification and Correlation*. Elsevier Scientific Software, Amsterdam.

K. R. Gabriel. 1971. The biplot graphic display of matrices with applications to principal component analysis. *Biometrika* 58, 3 (1971), 453–467.

Christophe Giraud. 2015. *Introduction to High-Dimensional Statistics*. CRC Press, Boca Raton, FL.

Joseph F. Hair, William C. Black, Barry J. Babin, Rolph E. Anderson, Ronald L. Tatham et al. 1998. *Multivariate Data Analysis*. Vol. 5. Prentice Hall, Upper Saddle River, NJ.

Haitao Zhao, Pong Chi Yuen, and J. T. Kwok. 2006. A novel incremental principal component analysis and its application for face recognition. *IEEE Trans. Syst. Man Cybernet., Part B (Cybernet.)* 36, 4 (Aug. 2006), 873–886. DOI: https://doi.org/10.1109/TSMCB.2006.870645

Simon Hales, Neil De Wet, John Maindonald, and Alistair Woodward. 2002. Potential effect of population and climate changes on global distribution of dengue fever: An empirical model. *Lancet* 360, 9336 (2002), 830–834.

David J. Hand. 2007. Principles of data mining. *Drug Safety* 30, 7 (2007), 621–622.

Lars Kai Hansen, Jan Larsen, Finn Årup Nielsen, Stephen C. Strother, Egill Rostrup, Robert Savoy, Nicholas Lange, John Sidtis, Claus Svarer, and Olaf B. Paulson. 1999. Generalizable patterns in neuroimaging: How many principal components? *NeuroImage* 9, 5 (1999), 534–544.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning (2nd ed.)*. Springer, New York, NY, USA.

Xiao Hu, Raj Subbu, Piero Bonissone, Hai Qiu, and Naresh Iyer. 2008. Multivariate anomaly detection in real-world industrial systems. In *Proceedings of the IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, 2766–2771.

Mia Hubert, Peter J. Rousseeuw, and Stefan Van Aelst. 2008. High-breakdown robust multivariate methods. *Stat. Sci.* 23, 1 (2008), 92–119.

J. Edward Jackson and Govind S. Mudholkar. 1979. Control procedures for residuals associated with principal component analysis. *Technometrics* 21, 3 (1979), 341–349.

Iain M. Johnstone and Arthur Y. Lu. 2009. On consistency and sparsity for principal component analysis in high dimensions. *J. Amer. Statist. Assoc.* 104 (2009), 682–693.

I. T. Jolliffe. 1986. *Principal Component Analysis*. Springer.

Juyang Weng, Yilu Zhang, and Wey-Shiuan Hwang. 2003. Candid covariance-free incremental principal component analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 8 (Aug. 2003), 1034–1040. DOI: https://doi.org/10.1109/TPAMI.2003.1217609

Kwang In Kim, Keechul Jung, and Hang Joon Kim. 2002. Face recognition using kernel principal component analysis. *IEEE Signal Process. Lett.* 9, 2 (2002), 40–42.

A. Levy and M. Lindenbaum. 1998. Sequential Karhunen-Loeve basis extraction and its application to images. In *Proceedings of the International Conference on Image (ICIP'98)*, Vol. 2. IEEE Comput. Soc, Chicago, IL, 456–460. DOI: https://doi.org/10.1109/ICIP.1998.723422

Max A. Little, Patrick E. McSharry, Stephen J. Roberts, Declan A. E. Costello, and Irene M. Moroz. 2007. Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *BioMed. Eng. OnLine* 6, 1 (2007), 23.

Chengjun Liu. 2004. Gabor-based kernel PCA with fractional power polynomial models for face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 5 (2004), 572–581.

Sanya Liu, Zhi Liu, Jianwen Sun, and Lin Liu. 2011. Application of synergetic neural network in online writeprint identification. *Int. J. Dig. Content Technol. Appl.* 5, 3 (2011), 126–135.

Haiping Lu, Konstantinos N. Plataniotis, and Anastasios N. Venetsanopoulos. 2008. MPCA: Multilinear principal component analysis of tensor objects. *IEEE Trans. Neural Netw.* 19, 1 (2008), 18–39.

Sawsan Mahmoud, Ahmad Lotfi, and Caroline Langensiepen. 2016. User activities outliers detection; integration of statistical and computational intelligence techniques. *Comput. Intell.* 32, 1 (2016), 49–71.

Sawsan M. Mahmoud, Ahmad Lotfi, and Caroline Langensiepen. 2012. User activities outlier detection system using principal component analysis and fuzzy rule-based system. In *Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments*. 1–8.

Majdi Mansouri, Mohamed Nounou, Hazem Nounou, and Nazmul Karim. 2016. Kernel PCA-based GLRT for nonlinear fault detection of chemical processes. *J. Loss Prevent. Process Industr.* 40 (2016), 334–347.

Leland McInnes, John Healy, and James Melville. 2018. UMAP: Uniform manifold approximation and projection for dimension reduction. Retrieved from https://arXiv:1802.03426.

Sebastian Mika, Bernhard Schölkopf, Alex J. Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. 1999. Kernel PCA and de-noising in feature spaces. In *Advances in Neural Information Processing Systems*. MIT Press, 536–542.

Boaz Nadler. 2008. Finite sample approximation results for principal component analysis: A matrix perturbation approach. *Ann. Stat.* 36 (2008), 2791–2817.

Debashis Paul. 2007. Asymptotics of sample eigenstructure for a large dimensional spiked covariance model. *Statistica Sinica* 17 (2007), 1617–1642.

Karl Pearson. 1895. Note on regression and inheritance in the case of two parents. *Proc. Roy. Soc. London* 58 (1895), 240–242.

Karl Pearson. 1901. On lines and planes of closest fit to systems of points in space. *London, Edinburgh, Dublin Philos. Mag. J. Sci.* 2, 11 (1901), 559–572.

Peter J. Rousseeuw, Michiel Debruyne, Sanne Engelen, and Mia Hubert. 2006. Robustness and outlier detection in chemometrics. *Crit. Rev. Anal. Chem.* 36, 3–4 (2006), 221–242.

Peter J. Rousseeuw and Mia Hubert. 2011. Robust statistics for outlier detection. *Wiley Interdisc. Rev.: Data Min. Knowl. Discovery* 1, 1 (2011), 73–79.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. 1997. Kernel principal component analysis. In *Proceedings of the International Conference on Artificial Neural Networks*. Springer, 583–588.

Bernhard Schölkopf, Alexander J. Smola, Francis Bach et al. 2002. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press.

Semeion. 2018. Dataset provided by Semeion, Research Center of Sciences of Communication, Via Sersale 117, 00128, Rome, Italy. Retrieved from www.semeion.it.

Vincent G. Sigillito, Simon P. Wing, Larrie V. Hutton, and Kile B. Baker. 1989. Classification of radar returns from the ionosphere using neural networks. *Johns Hopkins APL Tech. Digest* 10, 3 (1989), 262–266.

Pedro F. B. Silva, Andre R. S. Marcal, and Rubim M. Almeida da Silva. 2013. Evaluation of features for leaf discrimination. In *Proceedings of the International Conference Image Analysis and Recognition*. Springer, 197–204.

Jack W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes. 1988. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*. American Medical Informatics Association, 261.

Gilbert Strang. 1993. *Introduction to Linear Algebra*. Vol. 3. Wellesley-Cambridge Press, Wellesley, MA.

Alaa Tharwat. 2016. Principal component analysis-a tutorial. *Int. J. Appl. Pattern Recogn.* 3, 3 (2016), 197–240.

Michael E. Tipping and Christopher M. Bishop. 1999. Probabilistic principal component analysis. *J. Roy. Stat. Soc.: Ser. B (Stat. Methodol.)* 61, 3 (1999), 611–622.

Dhananjay Tomar, Yamuna Prasad, Manish K. Thakur, and Kanad K. Biswas. 2017. Feature selection using autoencoders. In *Proceedings of the International Conference on Machine Learning and Data Science (MLDS'17)*. IEEE, 56–60.

Matthew Turk and Alex Pentland. 1991. Eigenfaces for recognition. *J. Cogn. Neurosci.* 3, 1 (1991), 71–86.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9 (2008), 2579–2605.

Roman Vershynin. 2012. Introduction to the non-asymptotic analysis of random matrices. In *Compressed Sensing: Theory and Applications*. Cambridge University Press, Cambridge, UK, 210–268.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*. Association for Computing Machinery, New York, NY, 1096–1103.

Kyle W. Willett, Chris J. Lintott, Steven P. Bamford, Karen L. Masters, Brooke D. Simmons, Kevin R. V. Casteels, Edward M. Edmondson, Lucy F. Fortson, Sugata Kaviraj, William C. Keel et al. 2013. Galaxy Zoo 2: Detailed morphological classifications for 304 122 galaxies from the Sloan Digital Sky Survey. *Monthly Notices Roy. Astron. Soc.* 435, 4 (2013), 2835–2860.

Ian H. Witten, Eibe Frank, Mark A. Hall, and Christopher J. Pal. 2016. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometr. Intell. Lab. Syst.* 2, 1–3 (1987), 37–52.

Jian Yang, David Zhang, Alejandro F. Frangi, and Jing-yu Yang. 2004. Two-dimensional PCA: A new approach to appearance-based face representation and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 1 (2004), 131–137.

Jieping Ye, Ravi Janardan, and Qi Li. 2004. GPCA: An efficient dimension reduction scheme for image compression and retrieval. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 354–363.

I.-Cheng Yeh. 2006. Exploring concrete slump model using artificial neural networks. *J. Comput. Civil Eng.* 20, 3 (2006), 217–221.

Tianqi Yu, Xianbin Wang, and Abdallah Shami. 2017. Recursive principal component analysis-based data outlier detection and sensor data aggregation in IoT systems. *IEEE Internet Things J.* 4, 6 (2017), 2207–2216.

Hui Zou and Trevor Hastie. 2005. Regularization and variable selection via the elastic net. *J. Roy. Stat. Soc. Ser. B* 67 (2005), 301–320.

Hui Zou, Trevor Hastie, and Robert Tibshirani. 2006. Sparse principal component analysis. *J. Comput. Graph. Stat.* 15 (2006), 265–286.

Received August 2018; revised September 2020; accepted January 2021