Principal Component Analysis: Introduction, Derivation and Application

Seminar Data Mining

Christoph Waffler
School of Computation, Information and Technology
Technische Universität München
Email: christoph.waffler@tum.de

Abstract— **Principal Component Analysis (PCA) is a widely used technique in data science, known for its effectiveness in data reduction and analysis. In its more than 100-year history, PCA has established itself as a fundamental tool in the field of data science. In this paper, we address the mathematical foundations of PCA, derive its general formulation, and highlight the key properties that have contributed to its widespread use. Furthermore, we examine the wide range of applications in which PCA has been successfully applied. Finally, we briefly discuss the most commonly used extensions and alternatives to PCA.**

Keywords— **Principal Component Analysis, PCA, Data Analysis, Dimensionality Reduction, Feature Selection, Statistics, Regression, Data Mining**

I. Introduction

Through the proliferation of "Big Data" in the last decades, datasets with high-dimensional structure have become more and more common in many fields. For example in finance, gigabytes of online market data get accumulated daily due to the adoption of the Internet and e-commerce combined with increasing computing power [1, p. 1]. Also, in genetic experiments, it becomes feasible to capture the expression of thousands of genes from a single tissue [2, p. 303].

Therefore the question arises of how to deal with all those collected variables represented in the form of dimensions. By reducing the dimension of the feature space, we lower the number of relationships between variables to consider and make it easier to interpret the data. This is the goal of *dimension reduction* techniques.

One of the most popular dimension-reduction techniques is *principal component analysis* (PCA). It provides a powerful framework for extracting meaningful patterns and reducing the complexity of high-dimensional data. By identifying the most important features and components, PCA enables more efficient data representation, visualization, and analysis, leading to improved decision-making processes. Since PCA is applied directly to the data without any prior knowledge, it is considered an *unsupervised learning* method in the field of machine learning.

This paper aims to first provide the background for PCA (section II) and then explain the procedure of PCA (section III). Afterward, we will discuss the application scenarios (section IV) and drawbacks (section V) of PCA. Furthermore, we evaluate the benefits and limitations of PCA on a concrete example (section VI). Moreover, we will discuss related approaches to PCA (section VII) and finally, we will conclude with a summary of the key insights (section VIII).

II. Mathematical Background

As we start exploring principal component analysis, it's important to first understand the mathematical concepts behind it. This is because PCA is based on some key mathematical principles, and understanding these will help us understand PCA itself. In the next section, we start with the basics of matrix decomposition, then move on to variance and covariance and finish with the topic of the covariance matrix.

A. Matrix Decomposition

Given a square matrix $A \in \mathbb{R}^{n \times n}$, the eigenvalues λ and eigenvector v of A are defined as follows [3]:

$$A\,v = \lambda v \;\text{ with }\; v \neq 0 \;\text{ and }\; \lambda \in \mathbb{R} \tag{1}$$

We can transform the equation above into the following, where I_n is the identity matrix of the same size as A:

$$\begin{aligned}
&A\,v = \lambda v \;\text{ with }\; v \neq 0\\
&\Leftrightarrow (A - \lambda I_n)v = 0 \;\text{ with }\; v \neq 0\\
&\Leftrightarrow \det(A - \lambda I_n) = 0
\end{aligned}$$

The eigenvalues are the zeros of the characteristic polynomial χ_A of A:

$$\chi_A(\lambda) = \det(A - \lambda I_n). \tag{2}$$

We assume that the matrix A is diagonalizable, which means that there exists an invertible matrix $B \in \mathbb{R}^{n \times n}$ to calculate the diagonal form D of the matrix A [3]:

$$\begin{aligned}
D = B^{-1}\,A\,B &\Leftrightarrow A\,B = B\,D\\
&\Leftrightarrow (A\,b_1, \ldots, A\,b_n) = (\lambda_1 b_1, \ldots, \lambda_n b_n)\\
&\Leftrightarrow A\,b_i = \lambda_1 b_1, \ldots, A\,b_n = \lambda_n b_n.
\end{aligned} \tag{3}$$

The matrices A and D are matrix representations of the same linear mapping $f_A : \mathbb{R}^n \to \mathbb{R}^n$. The columns of the transformation matrix B are a basis of eigenvectors $b_1, \ldots, b_n$ of A to the eigenvalues $\lambda_1, \ldots, \lambda_n$, as shown in (3).

B. Variance and Covariance

After having delved into the concept of matrix decomposition, we will now explore the concepts of variance and covariance, which are two fundamental statistical concepts that play a significant role in the workings of PCA.

Variance and Covariance are important measures to understand the relationship and variability within a dataset. The variance $\mathrm{Var}[A]$ describes the average squared difference of a random variable A from its mean μ_A.

According to the definition of variance and linearity of expectation, the variance of two random variables A and B can be transformed to the following [4]:

$$\begin{aligned}&\mathrm{Var}[A+B]\\&=\mathbb{E}[A+B-\mathbb{E}[A+B]]^2\\&=\mathbb{E}[A-\mu_A+B-\mu_B]^2\\&=\mathbb{E}[A-\mu_A]^2+\mathbb{E}[B-\mu_B]^2+2\mathbb{E}[(A-\mu_A)(B-\mu_B)]\\&=\mathrm{Var}[A]+\mathrm{Var}[B]+2\cdot\mathbb{E}[(A-\mu_A)(B-\mu_B)] \qquad (4)\end{aligned}$$

This transformation shows that the calculation of the variance is not additive in contrast to the calculation of the expected value. The variance of two added variables is not only the sum of the variances of the single random variables, but also an additional term which is defined as the *covariance* $\mathrm{cov}(A, B)$ of the two random variables A and B [4]. It shows the *shared variability* between the two random variables and can be written with the means μ_X and μ_Y of X and Y as follows:

$$\mathrm{cov}(A,B) := \mathbb{E}[(A-\mu_A)(B-\mu_B)] \qquad (5)$$

C. The Covariance Matrix

After exploring variance and covariance, we will now proceed to examine the covariance matrix, which is another key component in the implementation of PCA.

For PCA, the covariance of a multidimensional dataset is of interest, instead of the covariance of two random variables.

So, given the vectors $\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}$ describing a sample of n random observations of dimension m, the dataset can be represented in a data matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$. For the purpose of PCA, we are primarily interested in the covariance of the columns $\boldsymbol{x_i}$ of $\boldsymbol{X}$, which we can be derived from the *sample covariance matrix* $\boldsymbol{S} \in \mathbb{R}^{m\times m}$, as described by [1, pp. 4-5]:

$$\mathrm{cov}(\boldsymbol{X}) = \boldsymbol{S} = \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{x_i}-\bar{x})^\top(\boldsymbol{x_i}-\bar{x}) \in \mathbb{R}^{m\times m}. \qquad (6)$$

Here, $\bar{x}$ is the *sample mean* and is defined by $\bar{x} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x_i} \in \mathbb{R}^{1\times m}$.

Moreover, we can also assume that the dataset is already centered around its mean, i.e., $\bar{x} = 0$; if it is not, we can simply subtract the mean from each observation to make the matrix centered. Because of this, the formula of the sample covariance matrix (6) can be simplified to [1, p. 11]

$$\mathrm{cov}(\boldsymbol{X}) = \boldsymbol{S} = \frac{1}{n-1}\sum_{i=1}^{n}\boldsymbol{x_i}^\top\boldsymbol{x_i} = \frac{1}{n-1}\boldsymbol{X}^\top\boldsymbol{X} \qquad (7)$$

It is important to point out that each element $\boldsymbol{S}[i, j]$ of the matrix $\boldsymbol{S}$ represents the covariance between the i-th and j-th variable of the dataset.

PCA aims to remove all correlations between the different variables, including the columns of $\boldsymbol{X}$. Consequently, the matrix $\boldsymbol{Y}$ resulting from the PCA should be uncorrelated. This means that the covariance matrix of $\boldsymbol{Y}$ should be a diagonal matrix, that solely includes the variances of each column individually.

III. Principal Component Analysis

After discussing the mathematical fundamentals that are relevant to PCA, we are now ready to introduce the principal component analysis itself, which brings all the previous mathematical concepts together.

In abstract terms, PCA identifies a specific linear transformation $\boldsymbol{P}$ such that the covariance matrix of the resultant matrix $\boldsymbol{Y} = \boldsymbol{XP}$ is a diagonal matrix, with diagonal entries in descending order.

We can transform the formula of the covariance matrix to the following:

$$\begin{aligned}\mathrm{cov}(\boldsymbol{Y}) &= \frac{1}{n-1}\boldsymbol{Y}^\top\boldsymbol{Y}\\&=\frac{1}{n-1}(\boldsymbol{XP})^\top(\boldsymbol{XP})\\&=\frac{1}{n-1}\boldsymbol{P}^\top\boldsymbol{X}^\top\boldsymbol{XP}\\&=\boldsymbol{P}^\top\mathrm{cov}(\boldsymbol{X})\,\boldsymbol{P} \qquad (8)\end{aligned}$$

As shown previously by (3), for a symmetric matrix $A \in \mathbb{R}^{m\times m}$, the eigenvalues λ_i and eigenvectors v_i can be found by the eigenvalue decomposition $D = B^\top A\,B$. Now, we set $A = \mathrm{cov}(\boldsymbol{X})$. It follows that $\boldsymbol{P} = B$ based on (8) and we get a diagonalization with

$$\mathrm{cov}(\boldsymbol{Y}) = \boldsymbol{P}^\top\mathrm{cov}(\boldsymbol{X})\boldsymbol{P} = D. \qquad (9)$$

In this formula $\boldsymbol{P} = B$ represents the transformation matrix, that is a basis of the eigenvectors of $\mathrm{cov}(\boldsymbol{X})$ to the eigenvalues $\lambda_1, \ldots, \lambda_n$.

Since PCA also requires the variables of $\boldsymbol{Y}$ to be in descending variance, $\boldsymbol{P}$ has to be chosen in such a way, that the eigenvalues are sorted in descending order to maximize the captured amount of explained variance in the data. Given that $\mathrm{cov}(\boldsymbol{Y})$ is obtained through an eigenvalue composition, it contains the eigenvalues of $\mathrm{cov}(\boldsymbol{X})$. Thus, we arrange the eigenvector in $\boldsymbol{P}$ to correspond with decreasing eigenvalues. Since the eigenvectors can be arbitrarily scaled by a constant factor, we normalize them to have a length of $\|\boldsymbol{p_i}\|_2 = 1$.

We refer to this ordered collection of normalized eigenvectors as the *principal components* $\boldsymbol{p_i}$ of $\boldsymbol{X}$, and construct $\boldsymbol{P}$ as follows:

$$\boldsymbol{P} = (\boldsymbol{p_1}, \boldsymbol{p_2}, \ldots, \boldsymbol{p_m}) \tag{10}$$

Now, due to the fact that the eigenvalues in $\boldsymbol{P}$ are sorted in descending order, we can only select the first k principal components and the first k eigenvalues to obtain a reduced dimensionality of k. Let $\boldsymbol{W}$ be a subset of $\boldsymbol{P}$ and contain those first $k < m$ principal components like $\boldsymbol{W} = (\boldsymbol{p_1}, \boldsymbol{p_2}, \ldots, \boldsymbol{p_k})$. This can now be applied in a projection $\boldsymbol{Y} = \boldsymbol{XW}$. After that, $\boldsymbol{Y}$ is a reduced dataset with k instead of m dimensions per sample.

IV. Application

A. Dimensionality Reduction

Principal component analysis (PCA) can play a critical role in feature selection [5]. Feature selection involves identifying the most informative subset of features from a larger set, while feature engineering involves creating new, derived features that capture essential characteristics of the data [6].

By projecting the original high-dimensional data onto a low-dimensional space defined by the principal components, PCA identifies the directions of maximum variance in the data. These principal components represent the most informative features that contribute significantly to the overall variability of the data set. Hence, we can effectively reduce the dimensionality of the data by selecting a subset of principal components while preserving the essential information. As a result, PCA is oftentimes used as a preprocessing step for other machine learning algorithms like deep learning models [5].

Furthermore, the dimensionality reduction can also be used to be able to visualize the data in a lower-dimensional space (cf. example in section VI).

B. Clustering

Often, clustering techniques are based on conventional metrics such as distance or density [7]. However, correlation-based approaches offer the advantage of uncovering more complex relationships between data points by using more advanced correlation models. PCA-based approaches start by applying PCA to extract a new set of uncorrelated dimensions. Subsequently, clusters are mined in this transformed space or its subspace (cf. section VI).

However, using PCA for clustering is sensitive to outliers which can lead to inaccurate results [8]. If a dataset contains a small fraction that doesn't correspond to the correct correlation structure, PCA is likely to be misled by this fraction or even fail completely. This can be mitigated by using *robust principal component analysis* (RPCA) as discussed in [8]. RPCA will be explained in subsection VII in more detail.

V. Limitations

A common criticism of PCA is its inherent complexity, which can make interpretation of the results difficult [5]. The principal components and projected vectors derived by the PCA are real-valued and can have both positive and negative values. The projected values are linear combinations of the features, while the principal components are linear combinations of the rows. In certain applications, such as a stock return analysis, as explained by [5], each factor is a linear combination of different time periods of stock returns. This property further complicates the interpretation of the learned factors. It becomes difficult to provide a simple explanation for the underlying reasons for the derived factors. As a consequence, it may be difficult for analysts to have full confidence in the results of PCA. If an analyst cannot clearly justify why they are investing in certain stocks based on the results of PCA, they are less likely to choose to use it.

Another drawback of PCA is that it is a linear method. This means it can only capture linear relationships between variables. If the underlying relationships are nonlinear, PCA may not be able to capture them. In such cases, nonlinear dimensionality reduction techniques such as kernel PCA may be more appropriate (cf. section VII).

Furthermore, the costs associated with the use of PCA are another negative point. Since PCA oftentimes uses the singular value decomposition (SVD) to compute the principal components (cf. section VII), it can be computationally expensive when processing large datasets [5]. To compute the SVD of a matrix with m features and n observations, the computational complexity is $\mathcal{O}(mn^2 + n^3)$ [9]. When we want only the k largest principal components, the computational complexity reduces to $\mathcal{O}(kn^2)$ [5]. However, this is still impractical for a large number of data points and features [5], [10].

VI. PCA for the Evaluation of Breast Cancer Samples

In this section, we explain how PCA can be used in genome-wide expression data to evaluate breast cancer samples. Since modern sequencing technologies can measure the expression of thousands of genes in a single experiment, the data is high-dimensional and thus difficult to interpret. PCA can be used to reduce the dimensionality of the data and make it easier to interpret.

Ringnér [2] used the GSE5325 dataset from the Gene Expression Omnibus database, which contains the expression of 27,648 genes in 105 breast cancer samples. This dataset is used in the following section. Using this concrete example in Python[1] with the help of the scikit-learn library [11], we can illustrate how PCA can represent samples in a lower-dimensional space to make it easier to interpret and visualize the data. Furthermore, we will test how PCA performs in the detection of clusters.

We will also include the estrogen receptor status of the samples in the analysis, since this is an important factor in breast cancer.

First, we focus on analyzing two genes, GATA3 and XBP1. We can graphically represent breast cancer samples based on the genes' expression profiles, which makes it easier to classify samples into estrogen receptor-positive (ER+) or negative

[1] The entire code can be found at https://github.com/chrissiwaffler/pca-breast-tumor

(ER-). Using PCA, we create new variables derived from the original variables. For this example, Figure 1 shows the first and second principal components represented as two axes in the plot. The first shows the direction with the highest sample variation, while the second shows the largest variation uncorrelated with the first.

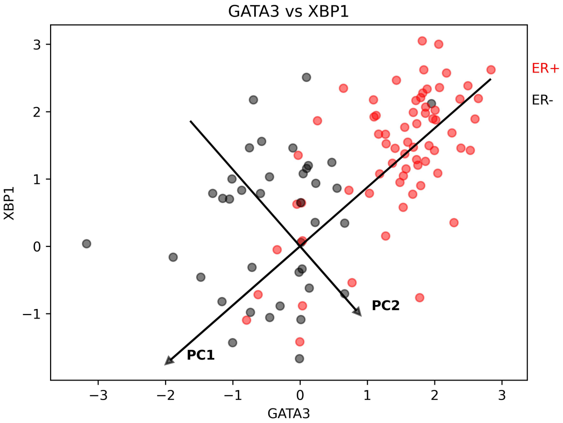

Fig. 1. Example of two genes plotted.

Furthermore, by projecting the data onto the first principal component, we can simplify our two-dimensional expression profiles into a single dimension as shown in Figure 2. This simplified model still distinguishes the samples based on their estrogen receptor status.

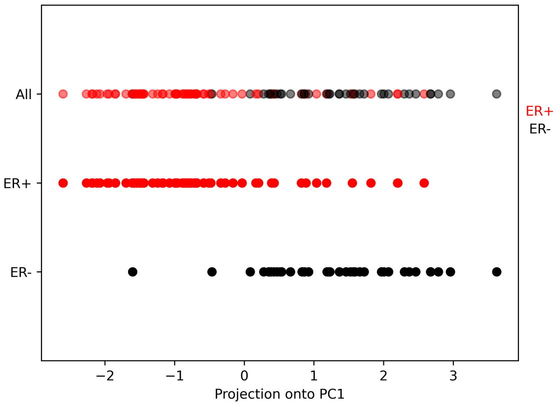

Fig. 2. Example of two genes plotted with the PCA axes.

Up until now, our application of PCA was demonstrated on two genes. Next, we expand this process to a much larger scale, handling a dataset comprising thousands of genes. We'll apply PCA to the 8,534 probes from the microarrays, each with expression measurements from 105 samples. To begin making sense of this larger data set's dimensionality, our initial step will involve examining the variance contained within each principal component for all genes (cf. Figure 3).

While the first few components contain more variance than the later ones, the first two components only explain about 21% of the variance. To retain 90% of the original variance, the first 63 components are needed. However, with 104 components, all of the original variance is retained, which is significantly much less than the initial 8,534 variables [2]. Here we can see, that when the variables outnumber the samples, PCA can effectively reduce the dimensionality to the maximum number of samples without any loss of information [12].

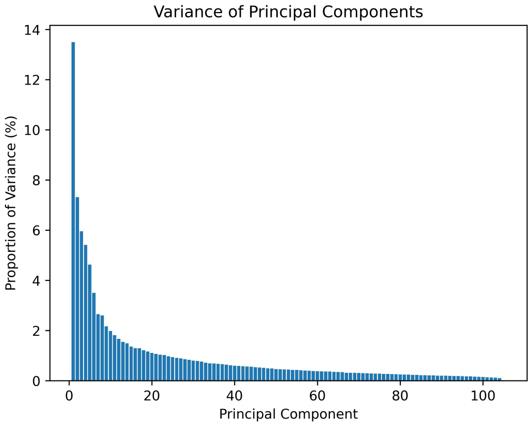

Fig. 3. Variance explained by PCA.

Besides that, the first two components of PCA may already hold crucial information about breast cancer samples. In Figure 4, we can see a PCA biplot where all samples are plotted in two dimensions using their projections onto the first two components and are colored according to their estrogen receptor status. Furthermore, two genes are plotted in blue using their weights for the components. We see the potential to reduce the data dimensionality from the number of genes to just two dimensions. The reduction however still maintains the ability to differentiate between ER+ and ER- samples.

The estrogen receptor status is known to significantly impact breast cancer cells' gene expression profiles. Although, it's crucial to note that PCA didn't form two distinct clusters in Figure 4. This example indicates that identifying unknown groups using PCA can be challenging.

Besides differentiating between the estrogen receptor gene, gene expression profiles can also distinguish breast cancer tumors based on whether they have gained DNA copies of the gene ERBB2 [2]. However, when this dataset is simplified to the first two principal components, as illustrated in Figure 5, this specific information is lost. This highlights that PCA's primary aim is to pinpoint directions with the most considerable variation, not necessarily those significant for classifying different groups.

In summary, this section described how PCA is used to interpret high-dimensional gene expression data from breast cancer. The GSE5325 dataset serves as an example, demonstrating how PCA simplifies multi-dimensional expression profiles into a single dimension while preserving crucial distinctions. However, it doesn't form distinct clusters and loses specific information when the dataset is reduced to the first two principal components. Advanced methods like robust

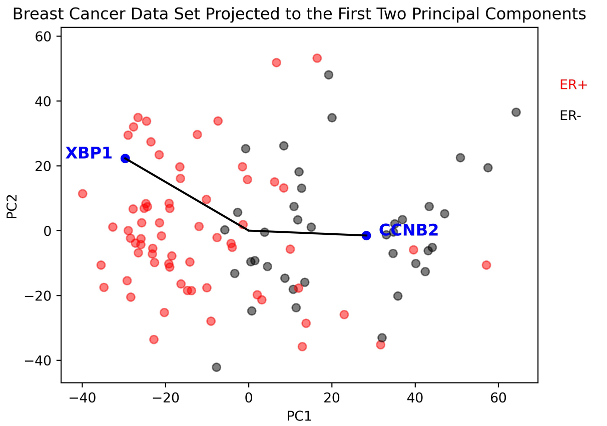

Fig. 4. PCA plot of the first two components. The scale is meant for the samples. For the genes XBP1 and CCNB2, marked in blue according to their weights, the scale should be divided by 950.

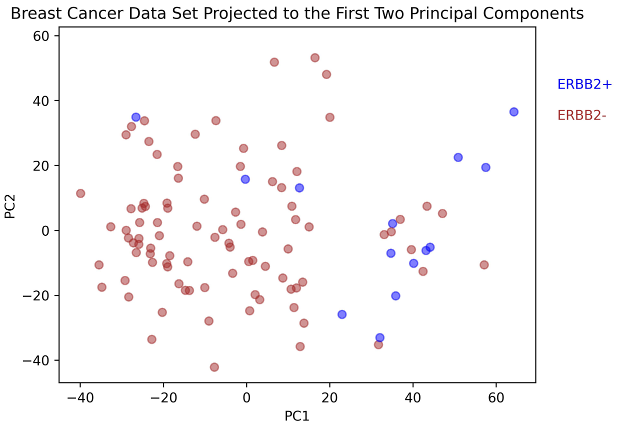

Fig. 5. PCA plot of the first two components. Similar to Figure 4, but with ERBB2 status instead of ER.

PCA (cf. section VII-B) or correlation-based clustering could overcome these limitations, offering improved outlier handling and group identification.

VII. Related Approaches

A. Singular Value Decomposition

In section III we derived PCA and its properties by using diagonalization of the covariance matrix of the dataset. However, in most frameworks such as sckit-learn PCA is implemented using the *singular value decomposition* (SVD) of the data matrix [13].

Given a data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times m}$ with n samples and m features, using SVD the matrix $\boldsymbol{X}$ will be decomposed into a product of three matrices $\boldsymbol{U}$, $\boldsymbol{\Sigma}$ and $\boldsymbol{V}^\top$ as shown in [3]:

$$\begin{aligned}&\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \\ &\quad\text{with } \boldsymbol{U} \in \mathbb{R}^{n \times n}, \boldsymbol{\Sigma} \in \mathbb{R}^{n \times m}, \boldsymbol{V} \in \mathbb{R}^{m \times m}. \qquad (11)\end{aligned}$$

Here, $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices, while $\boldsymbol{\Sigma}$ is a diagonal matrix. The diagonal entries of $\boldsymbol{\Sigma}$ contain the *singular values* and are sorted in descending order. Furthermore, the columns of $\boldsymbol{U}$ are called *left singular vectors*, and the columns of $\boldsymbol{V}$ are called *right singular vectors*.

According to [14] we can express the relation between the SVD and the covariance matrix $\boldsymbol{S}$ of $\boldsymbol{X}$. Since $\boldsymbol{S} = \frac{1}{n-1}\boldsymbol{X}^\top\boldsymbol{X}$ is symmetric and thus can be diagonalized to $\boldsymbol{S} = \boldsymbol{V}\boldsymbol{L}\boldsymbol{V}^\top$.

Here, $\boldsymbol{V}$ is a matrix containing the eigenvectors, and $\boldsymbol{L}$ is a diagonal matrix with the eigenvalues λ_i in decreasing order.

Furthermore, we can now show the relation between the SVD and the covariance matrix:

$$\begin{aligned}\boldsymbol{S} &= \frac{1}{n-1}\boldsymbol{X}^\top\boldsymbol{X} \\ &= \frac{1}{n-1}\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^\top\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top \\ &= \frac{1}{n-1}\boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^\top. \qquad (12)\end{aligned}$$

This shows us, that the right singular values of $\boldsymbol{V}$ are eigenvectors and the *singular values* s_i of $\boldsymbol{\Sigma}$ are related to the eigenvalues λ_i by the equation $\lambda_i = \frac{1}{n-1}s_i^2$. Finally, the principal components can now be calculated by $\boldsymbol{X}\boldsymbol{V} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{U}\boldsymbol{\Sigma}$ [15].

B. Extensions of PCA

PCA has been extended in various ways to enhance the usefulness of its results for exploratory data analysis. One of the most prominent extensions is *kernel principal component analysis* (KPCA), which allows for nonlinear dimensionality reduction. It uses *integral operator kernel functions* to map the data into a higher dimensional space, where the data is linearly separable [16].

Another extension is *sparse PCA*, which is a generalization of PCA that allows for a sparse representation of the data. This means that the resulting principal components are linear combinations of only a few of the original variables instead of all of them [17]. This is useful for exploratory data analysis because it allows for a more interpretable representation of the data.

Using the standard PCA, the presence of outliers can lead to a large error in the principal components (cf. Subsection V). Contrarily to that, robust PCA is an extension of PCA that is robust to outliers in the data [8]. It is based on the assumption that the data can be decomposed into a low-rank matrix and a sparse matrix. The low-rank matrix contains the principal components, while the sparse matrix contains the outliers [18].

C. Independent Component Analysis

PCA is limited by the assumption that the observed data were generated by a linear combination of independent sources, and thus may not be able to find the underlying components. This is where independent component analysis (ICA) comes into play [19].

Unlike PCA, which aims to minimize the loss resulted from the projection of the data onto the principal components, ICA looks for a decomposition of the data that minimizes the dependence between the basis vectors, which results in mutually independent components.

To make it more descriptive, consider the example of having two independent sources and two signals which are observed as a signal mixture [20]. Since these signals are from independent sources, we can assume that they are statistically independent. This conjecture allows the ICA to reverse the implication and make a practical but logically unwarranted assumption: If the statistically independent signals can be extracted from the signal mixture, then these extracted signals must stem from different sources and thus must be statistically independent.

So one can say that ICA aims to separate signal mixtures into statistically independent signals. If the assumption of statistical independence holds, each signal extracted by ICA represents a desired signal generated by a specific physical process.

VIII. Conclusion

In this paper, we presented an overview of principal component analysis by deriving it from the mathematical background such as *matrix decomposition* and *covariance*. In addition, we addressed the main applications of PCA and critically examine the main criticisms that have been raised against its use. Through detailed analysis, we improved our understanding of the theoretical foundations of PCA and its practical applications.

For those cases where PCA isn't the best choice, we reviewed some of the most popular alternatives and extensions. Notably, Independent Component Analysis is excellent for solving data separation tasks, while kernel PCA is a powerful tool for nonlinear dimensionality reduction. However, it should be noted that both ICA and KPCA have their own limitations, and the debate about their superiority over PCA is a highly controversial topic in literature [21], [22].

Overall, principal component analysis shows a strong suitability for regression analysis and dimensionality reduction. Consequently, it is expected to maintain its position as a fundamental component of statistical data analysis for the foreseeable future, albeit with possible modifications and further developments.

References

[1] J. Yao, S. Zheng, and Z. Bai, "Large sample covariance matrices and high-dimensional data analysis," *Cambridge UP, New York*, pp. 4–22, Apr. 2019.

[2] M. Ringnér, "What is principal component analysis?" *Nature Biotechnology*, vol. 26, no. 3, pp. 303–304, Mar. 2008. [Online]. Available: https://www.nature.com/articles/nbt0308-303

[3] C. Karpfinger, *Höhere Mathematik in Rezepten*, 4th ed. Springer Spektrum Berlin, 2022, pp. 429–435. [Online]. Available: https://link.springer.com/book/10.1007/978-3-662-54809-7

[4] N. Henze, *Stochastik für Einsteiger*, 6th ed. Springer, Mar. 2006, pp. 162–164. [Online]. Available: https://link.springer.com/book/10.1007/978-3-8348-9110-5

[5] A. Zheng and A. Casari, *Feature engineering for machine learning: principles and techniques for data scientists*, 1st ed. O'Reilly Media, Inc., 2018, pp. 99–113. [Online]. Available: https://learning.oreilly.com/library/view/feature-engineering-for/9781491953235/?ar=

[6] M. Kuhn and K. Johnson, *Applied Predictive Modeling*, 5th ed. Springer, 2016, pp. 35–40. [Online]. Available: https://link.springer.com/content/pdf/10.1007/978-1-4614-6849-3.pdf

[7] J. Han, M. Kamber, J. Pei, and M. Kaufmann, *Data Mining: Concepts and Techniques*, 3rd ed. Morgan Kaufmann, 2012, pp. 417–417.

[8] H.-P. Kriegel, E. Schubert, and A. Zimek, "A general framework for increasing the robustness of pca-based correlation clustering algorithms," in *20th Int. Conf. on Scientific and Statistical Database Management*, 2008.

[9] G. H. Golub and C. F. V. Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996, pp. 70–73.

[10] V. Vasudevan and M. Ramakrishna, "A hierarchical singular value decomposition algorithm for low rank matrices," pp. 2–4, 10 2019. [Online]. Available: http://arxiv.org/abs/1710.02812

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[12] "Lecture 7 — spectral methods," pp. 8–9, 2008. [Online]. Available: https://cseweb.ucsd.edu/~dasgupta/291-unsup/lec7.pdf

[13] "sklearn.decomposition.pca." [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

[14] amoeba (https://stats.stackexchange.com/users/28666/amoeba), "Relationship between svd and pca. how to use svd to perform pca?" Cross Validated, uRL:https://stats.stackexchange.com/q/134283 (version: 2023-02-22). [Online]. Available: https://stats.stackexchange.com/q/134283

[15] T. Hastie, R. Tibshirani, and J. H. Friedman, "The elements of statistical learning: Data mining, inference, and prediction," pp. 534–537, 2009. [Online]. Available: https://link.springer.com/book/10.1007/978-0-387-21606-5

[16] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *International conference on artificial neural networks*. Springer, 1997, pp. 583–588. [Online]. Available: https://people.eecs.berkeley.edu/~wainwrig/stat241b/scholkopf_kernel.pdf

[17] H. Zou, T. Hastie, and R. Tibshirani, "Sparse principal component analysis," *Journal of Computational and Graphical Statistics*, vol. 15, pp. 265–286, 2006. [Online]. Available: https://doi.org/10.1198/106186006X113430

[18] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, 6 2011. [Online]. Available: https://doi.org/10.1145/1970392.1970395

[19] P. Comon, "Independent component analysis, a new concept?" *Signal Processing*, vol. 36, pp. 287–314, 1994, higher Order Statistics. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0165168494900299

[20] J. V. Stone, "Independent component analysis," *A Bradford Book*, 2004.

[21] K. Baek, B. A. Draper, J. R. Beveridge, and K. She, "Pca vs. ica: A comparison on the feret data set," in *JCIS*, 2002, pp. 824–827. [Online]. Available: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=8a17e16de6b932ec42e269621e29d99e46591fef

[22] L. J. Cao and W. K. Chong, "Feature extraction in support vector machine: a comparison of pca, xpca and ica," in *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02.*, vol. 2, 2002, pp. 1001–1005 vol.2.