

A comprehensive review of Principal Component Analysis

Yubo Ye *

Department of Mathematics, University College London, UK

* Corresponding Author Email: Yubo.ye.24@ucl.ac.uk

Abstract. PCA (Principal Component Analysis) is a method aiming to reduce the dimensions among data analysis, with various applications in neurosciences, finance, and beyond. Data normalization, covariance matrix decomposition, eigenvalue-driven component selection, and other mathematical underpinnings of PCA will be methodically covered in this article. A comparison with SVD decomposition will also be made due to the similarities between the two methods. Additionally, we will discuss contemporary developments like sparse PCA, kernel PCA, and robust PCA that tackle nonlinearity and sparsity by integrating trends like PCA's integration with deep learning, the variation in applied circumstances, and its use in high-dimensional data presentation. Furthermore, this review will also highlight the inherent limits, such as nonlinearity issues, massive datasets, and data contamination. Throughout investigation, this review serves as a map for the researchers tackling with increasingly complex data environments requiring dimensionality reduction and are not certain with the specific PCA type selected to apply.

Keywords: Principal component analysis, Dimensionality Reduction, Correlation Variance, Data analysis, Kernel PCA.

1. Introduction

PCA was first introduced by Karl Pearson in 1901, is to reduce the dimensionality of a data set retaining as much as possible of the variation [1]. The PCA method could efficiently exclude redundant data and extract the most important information, which preserves two goals: preserving maximal variance in the data with the least dimensions as well as enabling the presentations of complex structures. Despite its simplicity, PCA has numerous extensions in addressing distinctive challenges. By using a nonlinear map to relate high-dimensional feature spaces to the input space, Kernel PCA utilizes kernel functions to compute the principal components in these spaces [2], provides a feasible way to nonlinear manifolds. Meanwhile, the sparse PCA uses the lasso (elastic net) to modify the principal components with sparse loadings [3]. Not to mention the robust PCA, where the goal of mitigating the sensitivity to outliers is achieved. As the datasets grow and complex, PCA encounters limitations such as linearity assumptions, scalability and interpretability. This review will conclude PCA's theoretical foundations and potential applications followed by some modern extensions and evaluations.

In following section, we will discuss contemporary developments like sparse PCA, kernel PCA, and robust PCA that tackle nonlinearity and sparsity by integrating trends like PCA's integration with deep learning, the variation in applied circumstances, and its use in high-dimensional data presentation. Furthermore, this review will also highlight the inherent limits, such as nonlinearity issues, massive datasets, and data contamination.

2. Mechanism of principle component analysis (PCA)

The PCA method, considering its nature of orthogonal transformation, could be achieved by linear transformation, which is a matrix multiplication as well. Among Euclidean space, the maximization of variance is executed by the algorithm combined with Singular value decomposition (SVD) method. For a dataset of given observations and features, our goal is to find a lower-dimensional subspace via an orthogonal transformation of the data matrix that maximizes the preserved variance. The relationships between the features are assumed to be linear, where directions with largest variance capturing the most structure [4].

Before constructing the covariance matrix, the idea of centering with mean value is required to be executed to eliminate the bias toward origin. After computing the covariance matrix, according to the Lagrange multipliers, under the constraints of unit vector, we could take partial derivative to the Lagrangian that could be simplified into an Eigenvalue equation. This method efficiently converts PCA into an eigenvalue problem, indicating that the principal components are eigenvectors of the covariance matrix. Singular value decomposition, outputting orthogonal matrices, performs among data centering in PCA method. The SVD method directly decomposes the matrix which avoids redundant covariance calculations. In summary, PCA is a special application of SVD on centered data [5].

3. Different types of pca method

3.1 Kernel PCA

The motivation of kernel PCA is to consider linearity since the kernel methods are a class of widely-used machine learning techniques that enable linear models—such as regression, SVMs, PCA, and LDA—to learn and represent complex nonlinear patterns [6]. Linear PCA may fail when data lies on nonlinear manifolds. It maps the data from high-dimensional space which is nonlinear to become linearly separable [7]. The core idea is to implement kernel function to compute the inner products by mapping the data to feature space. For kernels that depend on only dot products or distance in input space, KPCA's unitary invariance derives from the sole dependence of its algorithm on the kernel matrix's values [2]. KPCA works even if the feature space is of infinite dimension since the computations merely rely on the inner products [8]. Necessarily, multiple KPCA could be applied under high-dimensional space as well, where we could find an actual example in image features reduction [9]. KPCA also plays a significant role in deep learning since it retains following advantages: Solutions are from linear algebra that guarantees transparency, high efficiency for medium datasets as well as the customized kernels which ensure the portability. For instance, facial recognition would be a great example applying kernel PCA since variations like a smile in expressions are extremely non-linear [10], where we could find a solution among Wang & Zhang's research (2010) that stated the experimental results of KPCA outperforms conventional PCA both in dimensionality reduction and overall performance, achieving up to 90% accuracy in their experiments [11].

3.2 Sparse PCA

When analyzing datasets with thousands of features, standard PCA produces dense loadings because every feature may occupy a weight in each component. The sparse PCA, integrating the sparse constraints into the optimization process, could continue in finding out the maximum variance. The Lasso method applies an L1 penalty (a bound on the sum of the absolute coefficient values) to the regression model, resulting in a sparse solution where some coefficients are driven to zero [4], could continuously shrink the coefficients toward zero, thus gaining its prediction accuracy via the bias variance trade-off could produce a sparse model. This is achieved by using an L1-norm (Lasso) penalty. To implement SPCA, we directly incorporate L1 penalties into the singular value decomposition (SVD) form underlying PCA or reconstruct it in a regression form [3]. SPCA imposes a requirement: the loadings vector for each principal component should contain mostly zeros. By setting many loadings to zero, SPCA automatically identifies a small subset of relevant features associated with each component which transforms the component from an abstract combination into an exact pattern. Its ability to extract meaningful and sparse patterns from complex datasets has rendered SPCA as an essential tool in data analysis. Among all applications, Genomics would be the most classic ones as there are more features than samples, one of whose solutions using SPCA obtains higher accuracy and relevancy to recognize the genes by replacing the elastic net with $L_{2,1}$ -norm penalty, which strongly supports the statement [12].

3.3 Robust PCA

Though the methods above are relatively comprehensive, the real-world datasets are usually contaminated by mess such as corrupted entries and missing value. Under the assumption of Gaussian errors, PCA is proven to be the optimal method for obtaining a low-rank approximation of high-dimensional data in terms of minimizing least-squares error [7]. The Robust Principal Component Analysis (RPCA) method reimagines the data matrix as the sum of two distinct components called low-rank matrix and sparse matrix. The former matrix represents the data captured by PCA under ideal conditions and the latter matrix is assumed to be sparse, meaning most of its entries are zero. An efficient recovery of most low-rank matrices from most error sign-and-support patterns via a simple convex program was established by Wright, Ganesh, Rao, Peng, and Ma [5], which is also an exact template.

RPCA's ability to separate signal from corruption renders its wide use in areas like face recognition, medical imaging and financial model etc. Yuan, Xie, Ma, and Lam (2013) suggested color facial denoising based on the RPCA technique, which indicates that their approach can attain a greater expectation in terms of both qualitative evaluation and visual quality than KPCA [13], in contrast to earlier facial recognition using the KPCA method. Where classical PCA falls short, RPCA offers a strong foundation for high computing needs. An approach that is not overly sensitive to any one piece of data is considered robust. Next, the concept of a norm of projection is presented [6]. Wang and Zhu also proposed that PCA is a package of SVD. In addition, we can get the two directions of PCA by using SVD, if we decompose the eigenvalue of matrix we can only get PCA with one direction [14].

4. Conclusion

Principal Component Analysis is widely recognized as one of the most prominent data analysis approaches. Its strength comes from a straightforward yet efficient method: breaking down high-dimensional, frequently correlated variables into a collection of orthogonal parts that represent the directions of greatest variation.

However, the history of PCA demonstrates both its flexibility and the difficulty in addressing its limits in the face of data issues. High-dimensional datasets are subjected to sparse PCA. By enforcing sparsity constraints on component loadings, SPCA utilizes an amount of explained variance to gain clarity, transforming abstract principal components into interpretable factors defined by a small subset of meaningful features.

Simultaneously, Robust PCA emerged to conquer PCA's weakness in handling corruptions. By decomposing data into a low-rank matrix representing the true underlying structure and a sparse matrix capturing anomalies, RPCA provides a practical framework for reliable analysis in noisy, real-world environments. Its ability to identify and avoid corruption is undeniably strong.

Throughout investigation, this review serves as a map for the researchers tackling with increasingly complex data environments. In summary, PCA remains as relevant as ever. Evolving from a classical method for reducing Gaussian noise, it now encompasses robust variants such as SPCA and RPCA, highlighting its continued importance. Traditional PCA still serves as a cornerstone for analyzing standardized datasets, while SPCA and RPCA address the challenges posed by high-dimensional data and realistic noisy environments. All three methods are effective for dimensionality reduction. The selection among them should be guided by the nature of the data and the purpose of the analysis: whether the aim is to explain maximum variance (PCA), improve interpretability (SPCA), or withstand data corruption (RPCA). As datasets become larger and more complex, the fundamental ideas behind PCA—augmented by modern adaptations for robustness and sparsity—will continue to play a vital role in identifying meaningful patterns across diverse types of information.

References

- [1] Jolliffe, I. T. (2002). *Principal component analysis* (2nd ed.). Springer-Verlag.
- [2] Schölkopf, B., Smola, A., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5), 1299–1319.
- [3] Zou, H., Hastie, T., & Tibshirani, R. (2004). Sparse principal component analysis [Unpublished manuscript]. Department of Statistics, Stanford University.
- [4] Jolliffe, I. T., Trendafilov, N. T., & Uddin, M. (2003). A modified principal component technique based on the lasso. *Journal of Computational and Graphical Statistics*, 12(3), 531–547.
- [5] Wright, J., Ganesh, A., Rao, S., Peng, Y., & Ma, Y. (2009). Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems* (Vol. 22, pp. 2080–2088). MIT Press.
- [6] Candès, E. J., Li, X., Ma, Y., & Wright, J. (2011). Robust principal component analysis? *Journal of the ACM*, 58(3), Article 11.
- [7] Bao, B.-K., Liu, G., Xu, C., & Yan, S. (2012). Inductive robust principal component analysis. *IEEE Transactions on Image Processing*, 21(8), 3794–3800.
- [8] Peng, C., Chen, Y., Kang, Z., Chen, C., & Cheng, Q. (2020). Robust principal component analysis: A factorization-based approach with linear complexity. *Information Sciences*, 513, 581–599.
- [9] Ge, W., Hongzhe, X., Weibin, Z., Weilu, Z., & Baiyang, F. (2011, April). Multi-kernel PCA based high-dimensional images feature reduction. In *2011 International Conference on Electric Information and Control Engineering* (pp. 5966–5969).
- [10] Cheng, P., Li, W., & Ogunbona, P. (2009, December). Greedy approximation of kernel PCA by minimizing the mapping error. In *2009 Digital Image Computing: Techniques and Applications* (pp. 303–308).
- [11] Wang, Y., & Zhang, Y. (2010). Facial recognition is based on kernel PCA. *2010 Third International Conference on Intelligent Networks and Intelligent Systems* (pp. 88-91). IEEE.
- [12] Lu, Y., Gao, Y.-L., Liu, J.-X., Wen, C.-G., Wang, Y.-X., & Yu, J. (2016). Characteristic gene selection via L_{2,1}-norm sparse principal component analysis. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (pp. 1828–1833).
- [13] Yuan, Z., Xie, X., Ma, X., & Lam, K.-M. (2013). Color facial image denoising based on RPCA and noisy pixel detection. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 2449-2453). IEEE.
- [14] Wang, Y., & Zhu, L. (2017). Research and implementation of SVD in machine learning. In **2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)** (pp. 471-475). IEEE.