Principal Component Analysis Tutorial

September 17, 2025

1 Introduction

Principal Component Analysis (PCA) seeks orthogonal directions that capture maximal variance, providing dimensionality reduction, visualization, and noise suppression for numerical data. By projecting observations onto a low-dimensional subspace spanned by top principal components, PCA yields compact representations while preserving dominant structure.

2 Theory and Formulas

2.1 Covariance Matrix and Eigendecomposition

Given centered data matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, the empirical covariance is

$$\mathbf{S} = \frac{1}{n-1}\mathbf{X}^\top \mathbf{X}. \tag{1}$$

PCA solves the eigenvalue problem $\mathbf{S}\mathbf{u}_k = \lambda_k \mathbf{u}_k$ with $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$. The k-dimensional principal subspace is spanned by $\mathbf{U}_k = [\mathbf{u}_1, \dots, \mathbf{u}_k]$.

2.2 Projection and Reconstruction

Projected scores (principal components) are given by

$$\mathbf{Z} = \mathbf{X}\mathbf{U}_k, \tag{2}$$

while the rank-k reconstruction is $\hat{\mathbf{X}} = \mathbf{Z}\mathbf{U}_k^\top$. The fraction of variance explained by the first k components equals

$$\mathrm{ExplainedVariance}(k) = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{j=1}^{d} \lambda_j}. \tag{3}$$

2.3 Singular Value Decomposition View

PCA can also be expressed through SVD: $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$. Columns of $\mathbf{V}$ are eigenvectors of $\mathbf{S}$; singular values satisfy $\sigma_i^2 = (n-1)\lambda_i$. This perspective supports efficient computation for high-dimensional data.

3 Applications and Tips

- **Visualization**: project high-dimensional data to two or three principal components to reveal clusters or trends.
- **Preprocessing**: reduce dimensionality before clustering or regression to mitigate multicollinearity and noise.
- **Compression**: store only principal scores and loadings for recommender systems or image compression pipelines.
- **Best practices**: center features, optionally scale to unit variance, inspect explained variance curves, and monitor for component flipping when interpreting axes.

4 Python Practice

The script `gen_pca_figures.py` generates a synthetic dataset with correlated features, fits PCA, and saves both a projection plot and an explained-variance curve.

Listing 1: Excerpt from gen$_{p}ca_{f}igures.py$

```
from sklearn.decomposition import PCA

pca = PCA(n_components=3, whiten=False, random_state=7)
pca.fit(points)
projected = pca.transform(points)

explained = np.cumsum(pca.explained_variance_ratio_)
```

5 Result

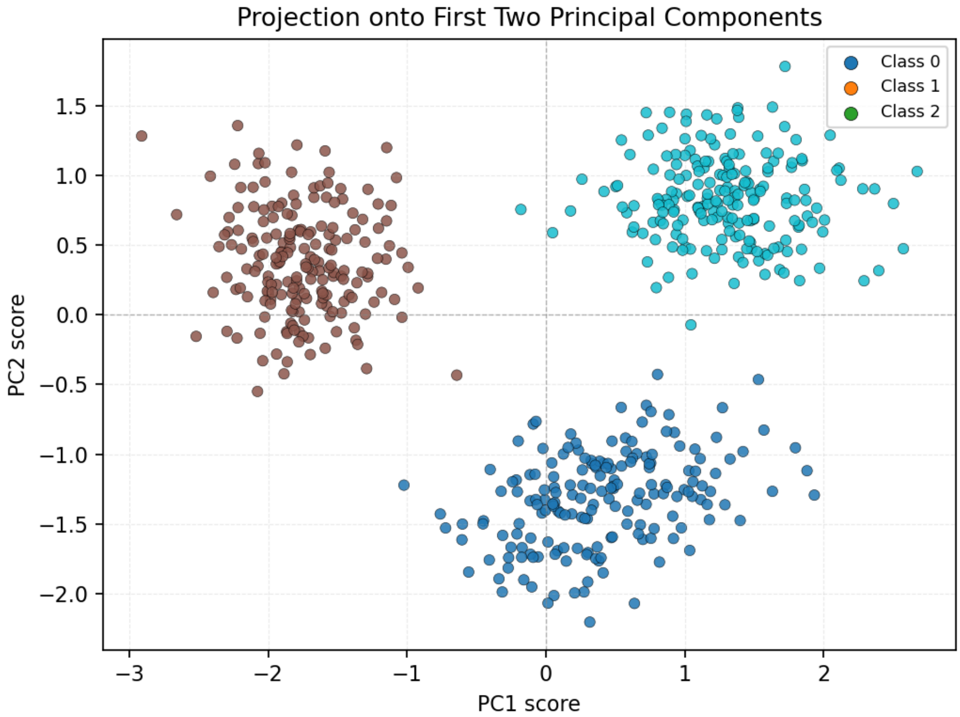

Figure 1: Scatter of the first two principal components with class colors

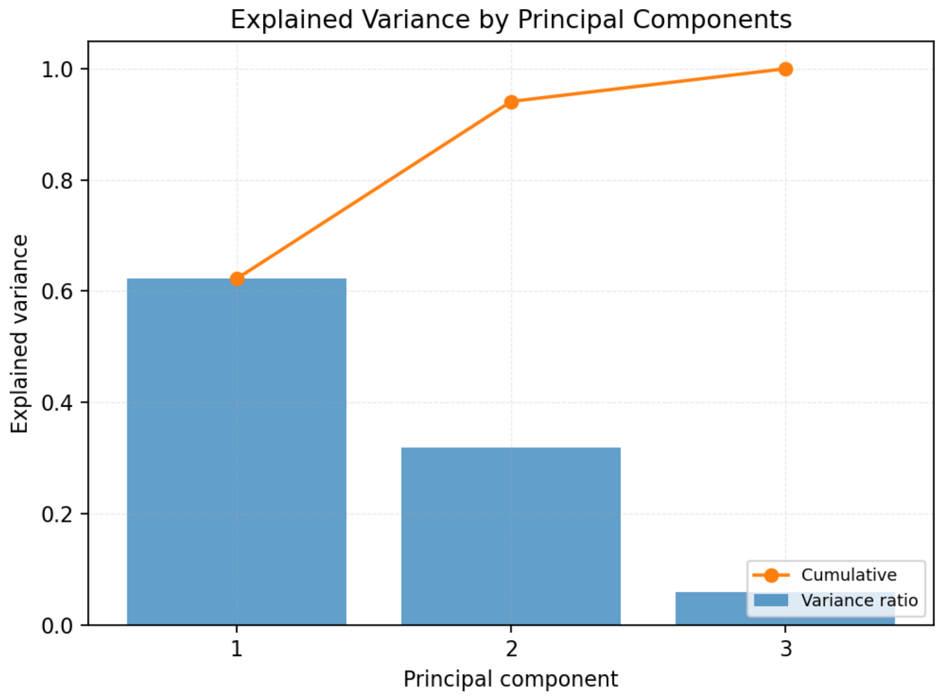

Figure 2: Explained variance ratio and cumulative curve across components

6 Summary

PCA extracts orthogonal directions of maximal variance via eigenvalue decomposition or SVD. Low-dimensional projections and explained variance diagnostics enable practitioners to balance compression with information retention. The synthetic example illustrates how scatter plots of principal scores and variance curves guide component selection.