EMMA ding

emmading.com

info@datainterviewpro.com

Principle Components Analysis (PCA)

Lesson Structure

What Is PCA?
How PCA Works
Pros and Cons

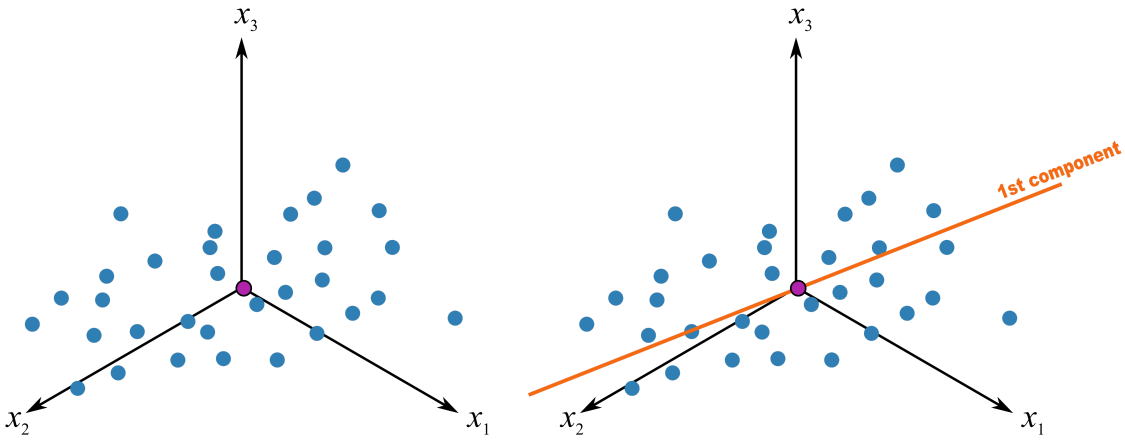
📌 Interview Questions

- What is principal component analysis? How does it work? Explain the sort of problems you would use PCA for.
- Describe PCA's formulation and derivation in matrix form.
- What are the pros and cons of PCA? Explain its limitations as a method.

▼ What Is PCA?

PCA is a **dimensionality reduction** technique that transforms input features into their principal components. It converts a set of observations of possibly correlated features into a set of values of **linearly uncorrelated** features.

- **Goal**: map the data from the original high-dimensional space to a lower-dimensional space that captures as much of the **variation** in the data as possible. It aims to find the most useful subset of dimensions to summarize the data.
  - e.g. a dataset with 3 features → use PCA to extract the first principal component that captures the most variance in the data.



Credit: Kevin Dunn, Source: https://learnche.org/pid/contents

- **Linear** transformation: PCA finds a sequence of linear combinations of features that have maximum variance and are uncorrelated.
- PCA is an **unsupervised** learning method: it does't use class labels.

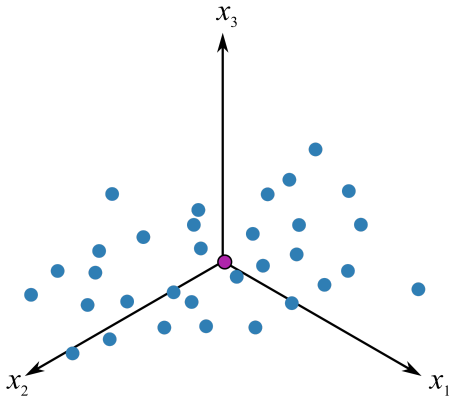
How PCA Works

General idea

Principal components are the directions of **maximum variance**, which has the effect of minimizing the information loss when you perform a projection or a compression down onto these principal components.

> ❓ Why does maximum variance mean minimum information loss?

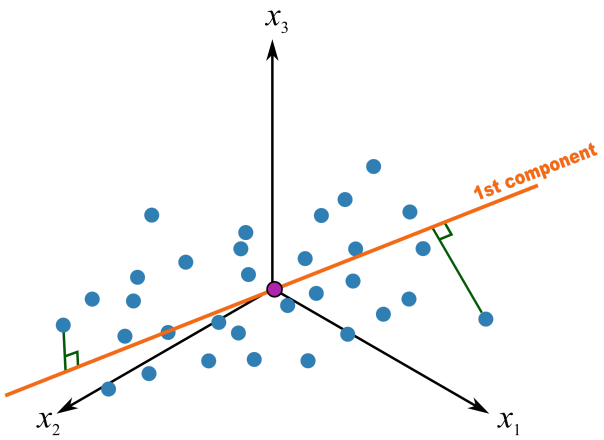
Support $x^{(i)}$ is an example in the original dataset, v is the principle component (a vector) and $a^{(i)}v$ is the transformed data point using PCA.



Minimize the mean squared error (MSE) between $x^{(i)}$ and $a^{(i)}v \rightarrow v$ would most closely transform $x^{(i)}$

$$min \sum_i (x^{(i)} - a^{(i)}v)^2$$

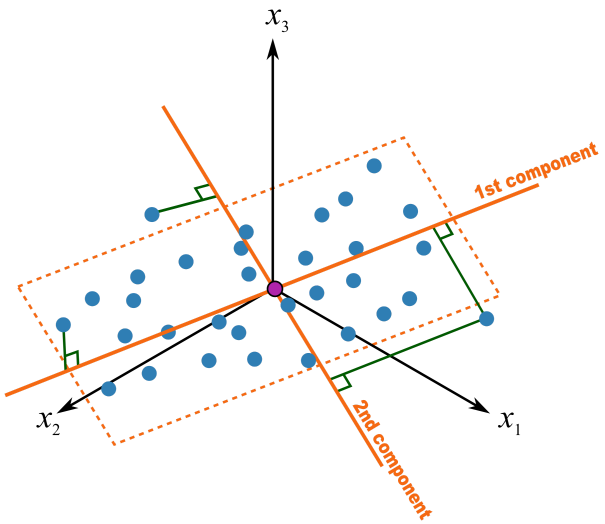
- $a^{(i)}$ (a scalar) can be calculated easily given v: it's the projection of $x^{(i)}$ onto v.
- Find v to minimize the residual variance
  - Minimize variance off $v \rightarrow v$ is the direction of maximum variance of the data.



Source: https://learnche.org/pid/contents

- Reduce the average of all the distances of every feature to the projection line (vector v), so it projects into the direction of maximal variance to minimize the distance from the original data to its newly transformed data → minimize the information loss.

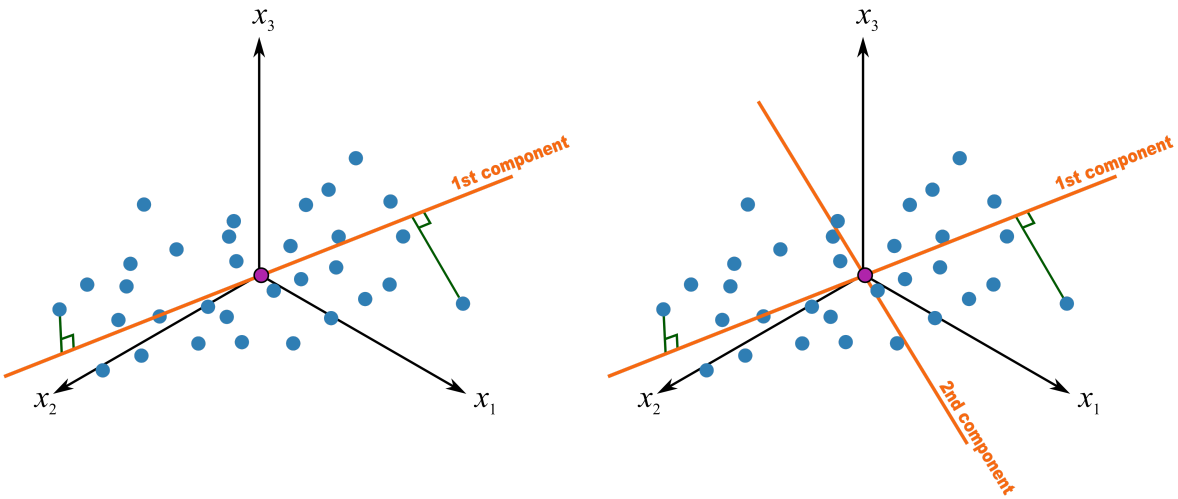
- For 2 PCs: $x^{(i)} = a^{(i)}v_1 + b^{(i)}v_2 + m$



Source: https://learnche.org/pid/contents

📍 All principal components are **uncorrelated (orthogonal)** to each other, so the 2nd principal component is mathematically guaranteed to not overlap with the 1st principal component.

- Even if the input features are correlated, the resulting PCs will be mutually uncorrelated → PCs can be treated as independent features.



Credit: Kevin Dunn, Source: https://learnche.org/pid/contents

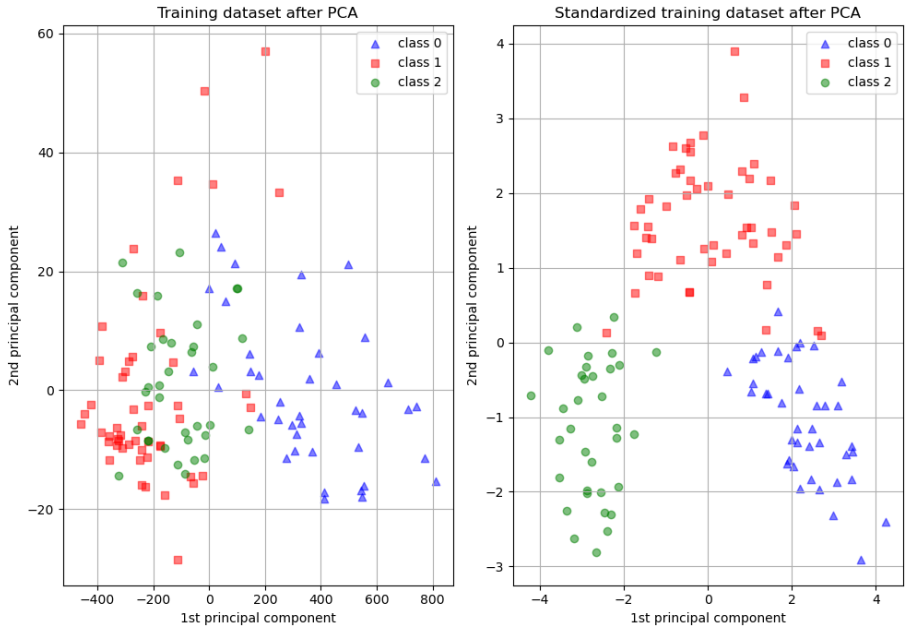
📌 Principle components are vectors that define a new coordinate system in which the nth axis goes in the direction of the nth highest variance of the data.

▼ Steps in PCA

▼ 1. Standardization

PCA is sensitive to the relative scaling of the original feature.

✏️ PCA are highly sensitive to data scaling, so we need to standardize the features prior to PCA if the features were measured on different units and assign equal importance to all features.



Source https://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html#

- Accuracy for the normal (i.e. unscaled) test dataset with PCA 81.48%
- Accuracy for the **standardized** test dataset with PCA 98.15%

> 📌 Standardization (or Z-score normalization) is an important preprocessing step for PCA.

▼ 2. Compute covariance matrix

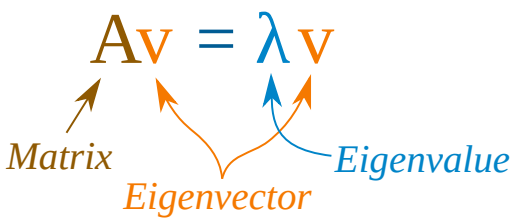
e.g. Covariance matrix $\sum$ of 3 features

$$\sum = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_2^2 & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 \end{bmatrix}$$

- Covariance matrix is a special case of a square matrix $A = A^T$.

▼ 3. Eigendecomposition

The factorization of a square matrix into eigenvectors and eigenvalues.



Decompose the covariance matrix $\sum$ into eigenpairs.

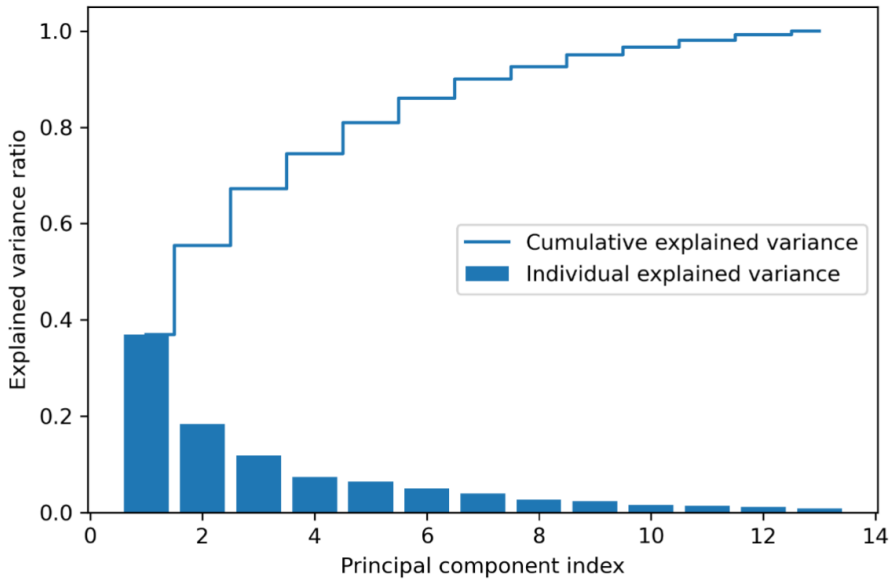
📌 Eigenvectors of the covariance matrix represent the principle components and the corresponding eigenvalues define their magnitude.

▼ **4. Choose k principal components ($k \leq d$)**

- Sort the eigenpairs in descending order of the eigenvalues
  - The eigenvector with the largest eigenvalue → the first principal component
  - The eigenvector with the second largest eigenvalue → the second principal component

▼ **How to select the number of principal components?**

1. To retain certain % of the variance, e.g. 90%.
2. Choose a cut off when it becomes apparent that adding more PCs doesn't get much more variance.
   - e.g. the first two PCs capture ~60% of the variance in the data.



The proportion of total variance explained by the principle components.

3. Specific use case, e.g. data visualization.

▼ **5. Feature transformation**

Transform d-dimensional feature spaces x to k-dimensional feature subspaces $x^{'}$.

$$x^{'} = xw$$

whereas w is a projection matrix constructed from the top k eigenvectors.

▼ Pros and Cons

▼ Advantages

- **Removes correlated features and noise in the data**
  - All the PCs are independent of each other. There is no correlation among them.
  - The first few PCs can capture majority of variance in the data and the rest just represent noise in the data.
  - A data preprocessing step before using a learning algorithm - transformed data are available to use.
- **Improves algorithm performance**
  - With high-dimensional features, the performance of an algorithm will degrade.
  - PCA speeds up the algorithm by getting rid of correlated features and noise which don't contribute in any decision making. The training time of the algorithms reduces significantly with less number of features.
- **Visualizes high-dimensional data**
  - PCA transforms a high dimensional data to low dimensional data (2 or 3 dimensions) so that the data can be visualized easily.

▼ Limitations

- PCA is not scale invariant, it is sensitive to the relative scaling of the input feature.
- Features become less interpretable**:** PCs are the linear combination of the original features and they are not as readable and interpretable as original features.
- Only based on the mean vector and covariance matrix. Some distributions (multivariate normal) are characterized by this but some are not.
- PCA is an unsupervised learning method, so it does not take labels into account.