Departamento de Métodos Cuantitativos

Unit 4. Dimensionality Reduction

Antonio Manuel Durán Rosal (amduran@uloyola.es)

Machine Learning - Ingeniería de Datos I

4º IITV - 3º ISW - 4º CVAD

Course 2025-2026

September 2025

Outline

Outline

1. Objectives
2. Introduction
3. Principal Component Analysis
4. t-distributed Stochastic Neighbour Embedding (t-SNE)

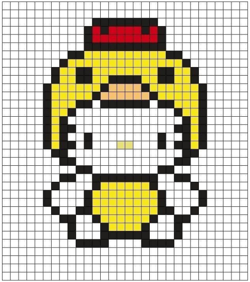
Objectives

- Understand the basic concepts of dimensionality reduction.
- Explain how PCA work.
- Apply PCA to datasets.
- Visualise reduced data in 2D or 3D.
- Evaluate the impact on machine learning models.
- Use Python (sklearn, matplotlib) to implement the techniques.

Outline

Dimensionality reduction

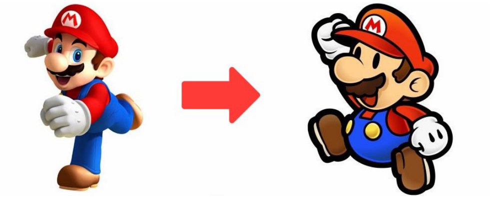
- **Dimensionality**: The number of features (variables) that describe each observation in a dataset.
  - Images: each pixel is a dimension.
  - Text: each unique word is a dimension.



	Discrete Nominal	Discrete Nominal	Binary Discrete Nominal	Ordinal Discrete Nominal	Continuous Ratio-scaled
Student ID	**Name**	**Course**	**Gender**	**Grades**	**Height (cm)**
S1	Alicent	Literature	Female	A	167.6
S2	Otto	Psychology	Male	C	185.9
S3	Criston	Computer Science	Male	B	179.8
S4	Laena	Life Science	Female	A+	161.5

Dimensionality reduction

- **Dimensionality reduction**: A process aimed at decreasing the number of features while preserving as much relevant information as possible.
- **Importance**:
  - Simplifies data analysis and visualisation.
  - Enhances the performance of machine learning models.
  - Reduces computational costs.



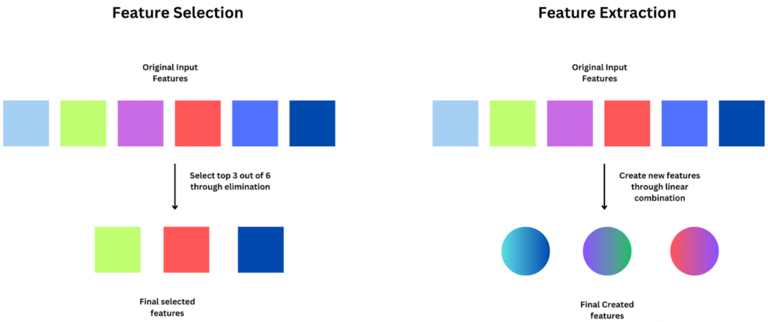
Problems of high dimensionality

- **The Curse of Dimensionality**: As the number of dimensions increases:
  - The distance between points becomes less meaningful.
  - Models require more data to generalise well.
- **Redundancy and Irrelevance**:
  - **Redundancy**: Variables that convey the same information.
  - **Irrelevance**: Variables that do not provide useful information.
- **Impact on ML Models**:
  - Increased risk of **overfitting**.
  - Longer training times.
  - Difficulty in identifying patterns.

How to reduce the number of variables?

- **Feature Selection**: Retains a subset of the original features.
  - **Filter**: Based on statistical metrics (correlation, chi-squared).
  - **Wrapper**: Uses predictive models to evaluate combinations.
  - **Embedded**: The model itself includes a feature selection mechanism (e.g. decision trees).
- **Feature Extraction**: Transforms the original data into a new set of features.
  - **PCA**: Combines features into new orthogonal variables.
  - **t-SNE**: Reduces dimensionality while preserving local proximity relationships.

How to reduce the number of variables?



Advantages of reduction

- Improves model efficiency.
  - Faster and more efficient models.
  - Reduced computational cost.
- Prevents overfitting.
  - Removing irrelevant features enhances generalisation ability.
- Facilitates visualisation.
  - More comprehensible representations (2D, 3D).
  - Useful for cluster analysis.
- Increases interpretability:
  - Fewer variables result in simpler, more explainable models.

Outline

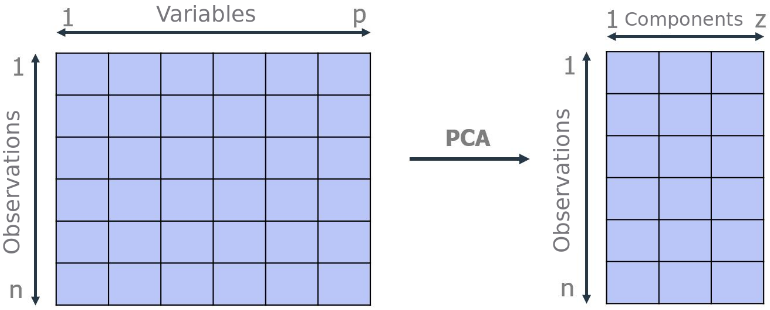
What is PCA?

- Principal Component Analysis (PCA) is a dimensionality reduction method.
- It allows for simplifying the complexity of high-dimensional spaces.
- It preserves as much information as possible while reducing dimensions.

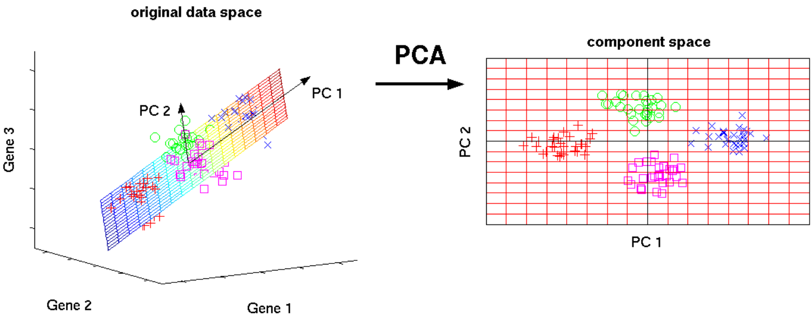
Dimensionality reduction with PCA

- Consider a sample with n individuals and p variables, that is, a sample space of p dimensions $(X_1, X_2, \ldots, X_p)$.
- PCA allows us to find a reduced number of underlying factors $z < p$ that explain approximately the same information as the original p variables.
- Whereas previously p values were needed to characterise each individual, now z values are sufficient.
- Each of these z new variables is called a **principal component**.

Dimensionality reduction with PCA



Dimensionality reduction with PCA



Applications and tools for PCA

- The PCA method allows information from multiple variables to be **"condensed"** into just a few principal components.
- It is important to remember that the values of the original variables are required to calculate these components.
- The main applications of PCA are:
  - **Visualisation** of data in lower-dimensional spaces.
  - **Preprocessing** of predictors before fitting supervised models.
- In Python, the `scikit-learn` library provides a class called `sklearn.decomposition.PCA` to implement PCA models.

Eigenvectors

- **Eigenvectors** represent a special case of matrix–vector multiplication.
- Example of matrix multiplication:

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 12 \\ 8 \end{pmatrix} = 4 \times \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

- The resulting vector is an **integer multiple** of the original vector.
- Therefore, the eigenvectors of a matrix are those vectors which, when multiplied by the matrix, result in the same vector or a scalar multiple of it.

Eigenvectors

We thus start from the following **fundamental equation**:

$$Av = \lambda v$$

Where:

- A: Square matrix.
- v: Eigenvector.
- λ: Eigenvalue.

Mathematical properties of eigenvectors

- Eigenvectors only exist for **square matrices**, and not all square matrices have eigenvectors.
- If a matrix is of size $n \times n$ and has eigenvectors, their number is n.
- When an **eigenvector is scaled** before being multiplied by the matrix, the result is a multiple of the same eigenvector:

$$A(\alpha v) = \alpha(Av) = \alpha\lambda v$$

  This happens because scaling a vector only changes its **length**, not its **direction**.
- All eigenvectors of a matrix are **perpendicular (orthogonal)** to each other, regardless of the number of dimensions they span.

Standardisation of eigenvectors

- Multiplying an eigenvector by a scalar only changes its **length**, not its nature.
- It is common to scale eigenvectors so that their **length is 1**, resulting in **standardised vectors** (unit norm).
- Example of standardisation:

$$\text{Original eigenvector: } \begin{pmatrix} 3 \\ 2 \end{pmatrix}$$

$$\text{Length: } \sqrt{3^2 + 2^2} = \sqrt{13}$$

$$\text{Standardised eigenvector: } \frac{1}{\sqrt{13}} \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} \frac{3}{\sqrt{13}} \\ \frac{2}{\sqrt{13}} \end{pmatrix}$$

- With this standardisation, all eigenvectors become **unit vectors**, making them easier to compare and use in further calculations.

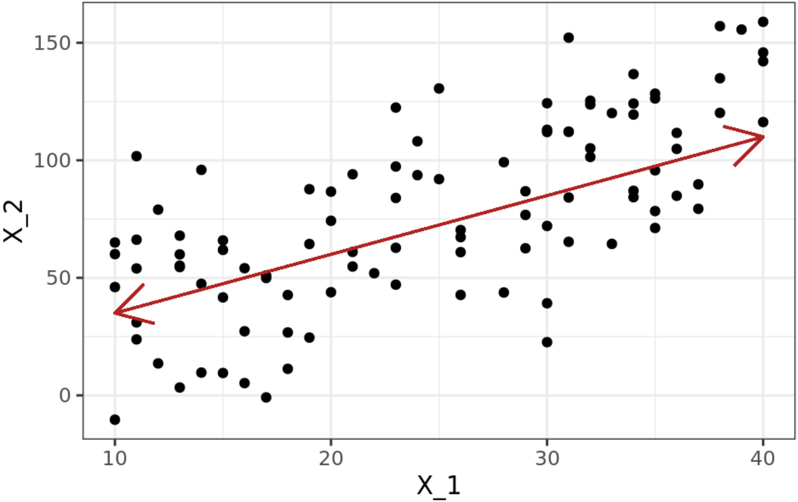
Eigenvalue

- When a matrix is multiplied by one of its eigenvectors, the result is a **multiple** of that same vector.
- The **number** by which the eigenvector is multiplied is known as the **eigenvalue**.
- Each eigenvector is associated with an eigenvalue, and vice versa.
- In the context of **PCA**:
  - Each **principal component** corresponds to an **eigenvector**.
  - The order of the principal components is determined by the **eigenvalues** in decreasing order.
  - The **first principal component** is the eigenvector with the **largest eigenvalue**, i.e., the one that explains the **most variance**.

Geometric interpretation

- An intuitive way to understand the **PCA** process is to interpret the principal components from a **geometric** perspective.
- Suppose we have a dataset with two variables (X_1, X_2).
- The **first principal component** Z_1 is defined as the **vector** that follows the direction of the **greatest variance** in the data (red line).
- The **projection** of each observation onto this direction is the **value of the first principal component**, known as the **Principal Component Score** z_{i1}.
- Graphically, the projection indicates how the observations are distributed along the direction that carries the most information.

First component



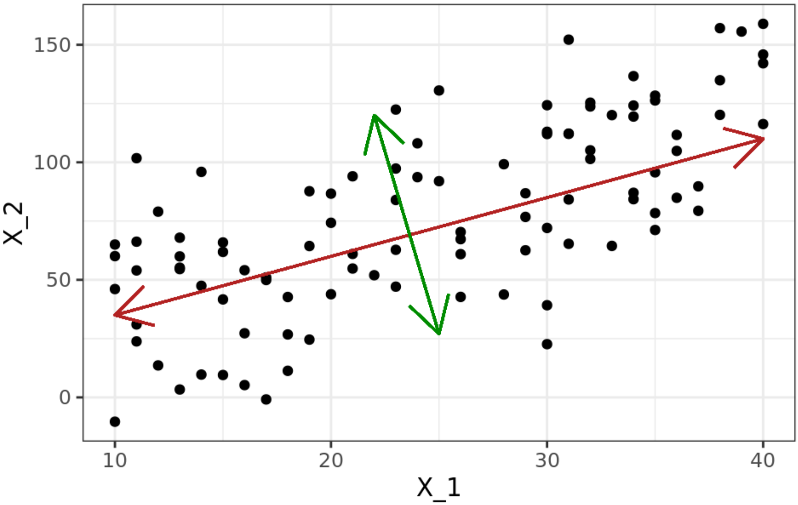
Second principal component

- The **second principal component** Z_2 represents the **second direction** in which the data shows the **greatest variance**.
- This second direction is **independent** of the first component, meaning it is **uncorrelated** with it.
- The **lack of correlation** between principal components implies that their directions are **perpendicular (orthogonal)**.
- Mathematically, **orthogonality** means that the **dot product** between the component directions is zero:

$$Z_1 \cdot Z_2 = 0$$

- In PCA, this orthogonality ensures that each component adds **new, non-redundant information**.

Second component



Principal components

- Each **principal component** Z_i is a **linear combination** of the **original variables**.
- The principal components can be interpreted as **new variables** formed by combining the original variables.
- The **first principal component** of a group of variables $(X_1, \dots, X_p)$ is the **normalised linear combination** that has the **greatest variance**:

$$Z_1 = \beta_{11}X_1 + \beta_{21}X_2 + \cdots + \beta_{p1}X_p$$

Where:

- β_{j1} are the **coefficients of the linear combination** (known as **loadings**).
- Each coefficient indicates the **contribution** of each original variable to the principal component.

Principal components

- The **normalisation condition** for the linear combination is:

$$\sum_{j=1}^{p} \beta_{j1}^2 = 1$$

  This normalisation ensures that the **length of the coefficient vector** is equal to **1**, allowing components to be compared on the same scale.

Procedure

1. Normalise the data.
2. Compute the covariance matrix.
3. Obtain eigenvectors and eigenvalues.
4. Sort by explained variance.
5. Reduce dimensionality.

Normalisation

- The **PCA** process identifies the **directions** in which the data shows the **greatest variance**.
- **Variance** is measured in the **squared units** of the variables.
- If the variables have **different scales**, those with **larger values** will **dominate** the analysis, masking the impact of variables with smaller scales.
- For this reason, it is **recommended** to **standardise** the data before applying PCA.

$$X_{\text{std}} = \frac{X - \overline{X}}{\sigma}$$

Where:

- $\overline{X}$ is the **mean** of the variable.
- σ is the **standard deviation**.

Computing the covariance matrix

- PCA solves an optimisation problem to find the values of the *loadings* that maximise the variance.
- The optimisation is performed by computing the eigenvectors and eigenvalues of the **covariance matrix** of the standardised data:

$$\Sigma = \frac{1}{n-1} X_{\text{std}}^{T} X_{\text{std}}$$

Computing eigenvectors and eigenvalues

- The **eigenvector** associated with the **largest eigenvalue** is the **first principal component**, as it is the direction that **maximises the variance**.
- The subsequent principal components are obtained from the **remaining eigenvectors**, sorted by **eigenvalues** in **descending** order.
- This method ensures that each component is **orthogonal** (uncorrelated) with the others.

Determinism in the PCA process

- The standard **PCA** process is **deterministic**, meaning that applying it to the same data always produces the **same principal components**.
- The **values of the loadings** (coefficients of the linear combinations) are always the same.
- The only possible difference is that the **sign of all loadings** may be inverted.
  - This is because the **loading vector** indicates the **direction** of the principal component.
  - A direction remains the same **regardless of sign**, as a component follows a **line** that extends in **both directions**.
- Similarly, the **Principal Component Scores** (component values for each observation) are **always the same**, except possibly for a change in **sign**.

PCA sensitivity to outliers

- The **PCA** method is **sensitive to outliers** because it relies on **variance**, which is affected by extreme values.
- It is important to **detect outliers** before applying PCA, especially in **multiple dimensions**, where unusual relationships are less evident.
- **Example:** A man who is **2 metres tall** and weighs **50 kg**:
  - **Individually**, neither value is extreme.
  - **Together**, it's an **atypical combination**.
- These **multidimensional outliers** can **distort** the **principal components** and affect the results of the **PCA**.

Proportion of explained variance

- In **PCA**, each principal component captures a **portion of the total variance** in the data.
- The **proportion of explained variance** indicates **how much information** from the original data is captured by each component.
- The proportion of variance explained by component i is:

$$\text{Proportion of Explained Variance (EVP)} = \frac{\lambda_i}{\sum_{j=1}^{p} \lambda_j}$$

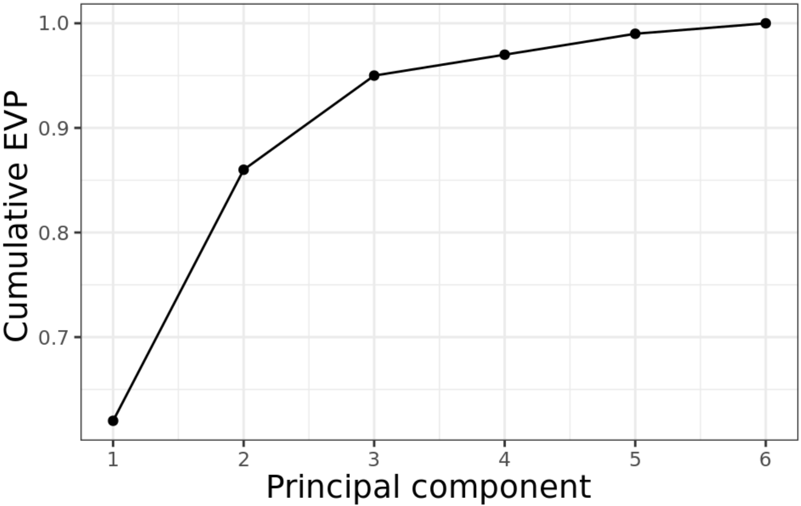
Optimal number of principal components

- This proportion helps decide **how many principal components** to retain in order to **summarise the data** without losing important information.
- The **cumulative EVP** is often used, which is the **sum** of the EVP explained by the first k components:

$$\text{Cumulative EVP} = \sum_{i=1}^{k} \frac{\lambda_i}{\sum_{j=1}^{p} \lambda_j}$$

- A common method is the **elbow plot**, which shows cumulative variance and helps identify the optimal number of components.

Optimal number of principal components



Outline

What is t-SNE?

- **t-SNE** is a dimensionality reduction technique particularly useful for **visualising complex data** in **2D or 3D**.
- Unlike **PCA**, which captures **global variance**, **t-SNE** preserves the **local relationships** between observations.
- Its goal is to project nearby points in the original space so that they **remain close** in the reduced space.
- The method is based on comparing **similarity probabilities** between points in high- and low-dimensional spaces.

References

- Benítez-Iglésias, R., Escudero-Bakx, G., Kanaan-Izquierdo, S., Masip-Rodó, D. (2014). Inteligencia artificial avanzada. Editorial UOC.
- Bishop, CM. (2006). Pattern recognition and machine learning. Springer.
- Géron, A. (2019). Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow (Second Edition). O'Reilly.
- James, G.; Witten, D.; R Hastie, T.; Tibshirani, R.; Taylor, J. (2023). An Introduction to Statistical Learning with Applications in Python. Springer.
- Mohri, M., Talwalkar, A., Rostamizadeh, A. (2019). Foundations of machine learning. Cambridge: The MIT Press.

¿Preguntas?