

Clase 9: Aprendizaje No Supervisado

Reducción de Dimensionalidad y Otros Métodos

Matías Leoni

Aprendizaje Automático I

Maestría en IA - Universidad de San Andrés

19 de Agosto de 2025

- **Parte 1: Fundamentos y PCA (45 min)**

- **Parte 1: Fundamentos y PCA (45 min)**
 - Motivación: El desafío de la alta dimensionalidad.

- **Parte 1: Fundamentos y PCA (45 min)**

- Motivación: El desafío de la alta dimensionalidad.
- La “Maldición de la Dimensionalidad”.

- **Parte 1: Fundamentos y PCA (45 min)**

- Motivación: El desafío de la alta dimensionalidad.
- La “Maldición de la Dimensionalidad”.
- Análisis de Componentes Principales (PCA): Teoría.

● **Parte 1: Fundamentos y PCA (45 min)**

- Motivación: El desafío de la alta dimensionalidad.
- La “Maldición de la Dimensionalidad”.
- Análisis de Componentes Principales (PCA): Teoría.
- Interpretación geométrica y matemática de PCA.

- **Parte 1: Fundamentos y PCA (45 min)**

- Motivación: El desafío de la alta dimensionalidad.
- La “Maldición de la Dimensionalidad”.
- Análisis de Componentes Principales (PCA): Teoría.
- Interpretación geométrica y matemática de PCA.

- **Pausa (15 min)**

- **Parte 1: Fundamentos y PCA (45 min)**
 - Motivación: El desafío de la alta dimensionalidad.
 - La “Maldición de la Dimensionalidad”.
 - Análisis de Componentes Principales (PCA): Teoría.
 - Interpretación geométrica y matemática de PCA.
- **Pausa (15 min)**
- **Parte 2: Aplicación de PCA y Métodos No Lineales (45 min)**

- **Parte 1: Fundamentos y PCA (45 min)**

- Motivación: El desafío de la alta dimensionalidad.
- La “Maldición de la Dimensionalidad”.
- Análisis de Componentes Principales (PCA): Teoría.
- Interpretación geométrica y matemática de PCA.

- **Pausa (15 min)**

- **Parte 2: Aplicación de PCA y Métodos No Lineales (45 min)**

- Varianza explicada y selección de componentes (Scree Plot).

- **Parte 1: Fundamentos y PCA (45 min)**

- Motivación: El desafío de la alta dimensionalidad.
- La “Maldición de la Dimensionalidad”.
- Análisis de Componentes Principales (PCA): Teoría.
- Interpretación geométrica y matemática de PCA.

- **Pausa (15 min)**

- **Parte 2: Aplicación de PCA y Métodos No Lineales (45 min)**

- Varianza explicada y selección de componentes (Scree Plot).
- Consideraciones prácticas de PCA: El escalado de datos.

- **Parte 1: Fundamentos y PCA (45 min)**

- Motivación: El desafío de la alta dimensionalidad.
- La “Maldición de la Dimensionalidad”.
- Análisis de Componentes Principales (PCA): Teoría.
- Interpretación geométrica y matemática de PCA.

- **Pausa (15 min)**

- **Parte 2: Aplicación de PCA y Métodos No Lineales (45 min)**

- Varianza explicada y selección de componentes (Scree Plot).
- Consideraciones prácticas de PCA: El escalado de datos.
- t-SNE para visualización de datos.

- **Parte 1: Fundamentos y PCA (45 min)**

- Motivación: El desafío de la alta dimensionalidad.
- La “Maldición de la Dimensionalidad”.
- Análisis de Componentes Principales (PCA): Teoría.
- Interpretación geométrica y matemática de PCA.

- **Pausa (15 min)**

- **Parte 2: Aplicación de PCA y Métodos No Lineales (45 min)**

- Varianza explicada y selección de componentes (Scree Plot).
- Consideraciones prácticas de PCA: El escalado de datos.
- t-SNE para visualización de datos.
- UMAP: una alternativa moderna a t-SNE.

- **Parte 1: Fundamentos y PCA (45 min)**

- Motivación: El desafío de la alta dimensionalidad.
- La “Maldición de la Dimensionalidad”.
- Análisis de Componentes Principales (PCA): Teoría.
- Interpretación geométrica y matemática de PCA.

- **Pausa (15 min)**

- **Parte 2: Aplicación de PCA y Métodos No Lineales (45 min)**

- Varianza explicada y selección de componentes (Scree Plot).
- Consideraciones prácticas de PCA: El escalado de datos.
- t-SNE para visualización de datos.
- UMAP: una alternativa moderna a t-SNE.

- **Bonus Track: Un Vistazo a Otros Métodos No Supervisados**

Motivación: ¿Por Qué Reducir la Dimensionalidad?

Motivación: ¿Por Qué Reducir la Dimensionalidad?

- Muchos datasets modernos tienen un gran número de características (p), a veces incluso más que observaciones ($p \gg n$).

Motivación: ¿Por Qué Reducir la Dimensionalidad?

- Muchos datasets modernos tienen un gran número de características (p), a veces incluso más que observaciones ($p \gg n$).
- **Ejemplos:** datos genómicos (miles de genes), procesamiento de imágenes (cada píxel es una característica), datos de texto (vocabulario extenso).

Motivación: ¿Por Qué Reducir la Dimensionalidad?

- Muchos datasets modernos tienen un gran número de características (p), a veces incluso más que observaciones ($p \gg n$).
- **Ejemplos:** datos genómicos (miles de genes), procesamiento de imágenes (cada píxel es una característica), datos de texto (vocabulario extenso).
- **Problemas de la alta dimensionalidad:**

Motivación: ¿Por Qué Reducir la Dimensionalidad?

- Muchos datasets modernos tienen un gran número de características (p), a veces incluso más que observaciones ($p \gg n$).
- **Ejemplos:** datos genómicos (miles de genes), procesamiento de imágenes (cada píxel es una característica), datos de texto (vocabulario extenso).
- **Problemas de la alta dimensionalidad:**
 - **Costo Computacional:** Más dimensiones implican mayor tiempo de procesamiento y más memoria.

Motivación: ¿Por Qué Reducir la Dimensionalidad?

- Muchos datasets modernos tienen un gran número de características (p), a veces incluso más que observaciones ($p \gg n$).
- **Ejemplos:** datos genómicos (miles de genes), procesamiento de imágenes (cada píxel es una característica), datos de texto (vocabulario extenso).
- **Problemas de la alta dimensionalidad:**
 - **Costo Computacional:** Más dimensiones implican mayor tiempo de procesamiento y más memoria.
 - **Sobreajuste (Overfitting):** Con muchas características, es más fácil que un modelo se ajuste al ruido en lugar de a la señal subyacente.

Motivación: ¿Por Qué Reducir la Dimensionalidad?

- Muchos datasets modernos tienen un gran número de características (p), a veces incluso más que observaciones ($p \gg n$).
- **Ejemplos:** datos genómicos (miles de genes), procesamiento de imágenes (cada píxel es una característica), datos de texto (vocabulario extenso).
- **Problemas de la alta dimensionalidad:**
 - **Costo Computacional:** Más dimensiones implican mayor tiempo de procesamiento y más memoria.
 - **Sobreajuste (Overfitting):** Con muchas características, es más fácil que un modelo se ajuste al ruido en lugar de a la señal subyacente.
 - **Redundancia y Ruido:** Muchas características pueden ser irrelevantes o estar altamente correlacionadas, añadiendo ruido.

La Maldición de la Dimensionalidad

La Maldición de la Dimensionalidad

- A medida que el número de dimensiones (p) aumenta, el volumen del espacio de características crece exponencialmente.

La Maldición de la Dimensionalidad

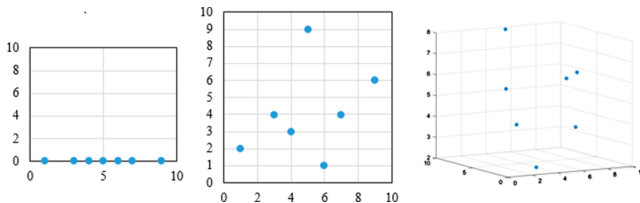
- A medida que el número de dimensiones (p) aumenta, el volumen del espacio de características crece exponencialmente.
- Los datos se vuelven **escasos (sparse)**. La distancia entre cualquier par de puntos tiende a ser similar, dificultando la tarea de algoritmos basados en distancia (como k-NN o clustering).

La Maldición de la Dimensionalidad

- A medida que el número de dimensiones (p) aumenta, el volumen del espacio de características crece exponencialmente.
- Los datos se vuelven **escasos (sparse)**. La distancia entre cualquier par de puntos tiende a ser similar, dificultando la tarea de algoritmos basados en distancia (como k-NN o clustering).
- Para mantener la misma densidad de datos, la cantidad de observaciones (n) necesarias crece de forma exponencial con p .

La Maldición de la Dimensionalidad

- A medida que el número de dimensiones (p) aumenta, el volumen del espacio de características crece exponencialmente.
- Los datos se vuelven **escasos (sparse)**. La distancia entre cualquier par de puntos tiende a ser similar, dificultando la tarea de algoritmos basados en distancia (como k-NN o clustering).
- Para mantener la misma densidad de datos, la cantidad de observaciones (n) necesarias crece de forma exponencial con p .
- La reducción de dimensionalidad busca proyectar los datos en un subespacio de menor dimensión, preservando la mayor cantidad de “información” posible.



Análisis de Componentes Principales (PCA)

Análisis de Componentes Principales (PCA)

- **Objetivo:** Encontrar una representación de baja dimensión de los datos que capture la mayor cantidad de varianza posible.

Análisis de Componentes Principales (PCA)

- **Objetivo:** Encontrar una representación de baja dimensión de los datos que capture la mayor cantidad de varianza posible.
- Es una técnica **lineal** y **no supervisada**.

Análisis de Componentes Principales (PCA)

- **Objetivo:** Encontrar una representación de baja dimensión de los datos que capture la mayor cantidad de varianza posible.
- Es una técnica **lineal** y **no supervisada**.
- PCA encuentra un nuevo conjunto de ejes (coordenadas) ortogonales, llamados **Componentes Principales (CP)**.

Análisis de Componentes Principales (PCA)

- **Objetivo:** Encontrar una representación de baja dimensión de los datos que capture la mayor cantidad de varianza posible.
- Es una técnica **lineal** y **no supervisada**.
- PCA encuentra un nuevo conjunto de ejes (coordenadas) ortogonales, llamados **Componentes Principales (CP)**.
- El primer componente principal (CP1) es la dirección en el espacio de características a lo largo de la cual los datos varían más.

Análisis de Componentes Principales (PCA)

- **Objetivo:** Encontrar una representación de baja dimensión de los datos que capture la mayor cantidad de varianza posible.
- Es una técnica **lineal** y **no supervisada**.
- PCA encuentra un nuevo conjunto de ejes (coordenadas) ortogonales, llamados **Componentes Principales (CP)**.
- El primer componente principal (CP1) es la dirección en el espacio de características a lo largo de la cual los datos varían más.
- El segundo componente principal (CP2) es la dirección, ortogonal a la primera, que captura la mayor varianza restante. Y así sucesivamente.

¿Qué son los Componentes Principales? (Matemáticamente)

¿Qué son los Componentes Principales? (Matemáticamente)

- Cada componente principal, Z_m , es una combinación lineal de las características originales X_1, X_2, \dots, X_p .

¿Qué son los Componentes Principales? (Matemáticamente)

- Cada componente principal, Z_m , es una combinación lineal de las características originales X_1, X_2, \dots, X_p .
- **Primer Componente Principal (Z_1):**

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \dots + \phi_{p1}X_p$$

¿Qué son los Componentes Principales? (Matemáticamente)

- Cada componente principal, Z_m , es una combinación lineal de las características originales X_1, X_2, \dots, X_p .
- **Primer Componente Principal (Z_1):**

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \dots + \phi_{p1}X_p$$

- Los valores ϕ_{j1} son los **loadings** (cargas) del primer componente. Forman el vector de loadings ϕ_1 .

¿Qué son los Componentes Principales? (Matemáticamente)

- Cada componente principal, Z_m , es una combinación lineal de las características originales X_1, X_2, \dots, X_p .

- **Primer Componente Principal (Z_1):**

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \dots + \phi_{p1}X_p$$

- Los valores ϕ_{j1} son los **loadings** (cargas) del primer componente. Forman el vector de loadings ϕ_1 .
- Los loadings se calculan para maximizar la varianza de Z_1 , sujeto a la restricción de que son normalizados:

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

¿Qué son los Componentes Principales? (Matemáticamente)

- Cada componente principal, Z_m , es una combinación lineal de las características originales X_1, X_2, \dots, X_p .

- **Primer Componente Principal (Z_1):**

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \dots + \phi_{p1}X_p$$

- Los valores ϕ_{j1} son los **loadings** (cargas) del primer componente. Forman el vector de loadings ϕ_1 .
- Los loadings se calculan para maximizar la varianza de Z_1 , sujeto a la restricción de que son normalizados:

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

- Los valores de Z_1 para cada observación son los **scores** o puntuaciones.

Interpretación Geométrica de PCA

Interpretación Geométrica de PCA

- PCA puede ser interpretado de dos maneras equivalentes:

Interpretación Geométrica de PCA

- PCA puede ser interpretado de dos maneras equivalentes:
- **1. Dirección de Máxima Varianza:**

Interpretación Geométrica de PCA

- PCA puede ser interpretado de dos maneras equivalentes:
- **1. Dirección de Máxima Varianza:**
 - El primer componente principal ϕ_1 define la línea que mejor captura la variabilidad de los datos.

Interpretación Geométrica de PCA

- PCA puede ser interpretado de dos maneras equivalentes:
- **1. Dirección de Máxima Varianza:**
 - El primer componente principal ϕ_1 define la línea que mejor captura la variabilidad de los datos.
 - Proyectar los datos sobre esta línea nos da los scores z_{i1} .

Interpretación Geométrica de PCA

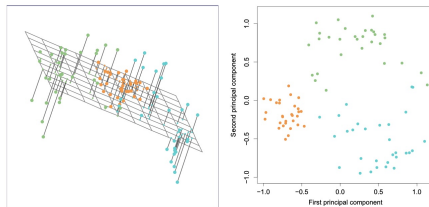
- PCA puede ser interpretado de dos maneras equivalentes:
- **1. Dirección de Máxima Varianza:**
 - El primer componente principal ϕ_1 define la línea que mejor captura la variabilidad de los datos.
 - Proyectar los datos sobre esta línea nos da los scores z_{i1} .
- **2. Subespacio de Mínima Distancia:**

Interpretación Geométrica de PCA

- PCA puede ser interpretado de dos maneras equivalentes:
- **1. Dirección de Máxima Varianza:**
 - El primer componente principal ϕ_1 define la línea que mejor captura la variabilidad de los datos.
 - Proyectar los datos sobre esta línea nos da los scores z_{i1} .
- **2. Subespacio de Mínima Distancia:**
 - El plano (o hiperplano) definido por los primeros M componentes principales es el que está más cerca de las observaciones.

Interpretación Geométrica de PCA

- PCA puede ser interpretado de dos maneras equivalentes:
- **1. Dirección de Máxima Varianza:**
 - El primer componente principal ϕ_1 define la línea que mejor captura la variabilidad de los datos.
 - Proyectar los datos sobre esta línea nos da los scores z_{i1} .
- **2. Subespacio de Mínima Distancia:**
 - El plano (o hiperplano) definido por los primeros M componentes principales es el que está más cerca de las observaciones.
 - Minimiza el error de reconstrucción (la suma de distancias euclidianas al cuadrado desde los puntos al plano).



Cálculo de los Componentes Principales

Cálculo de los Componentes Principales

- El proceso para calcular los componentes principales es:

Cálculo de los Componentes Principales

- El proceso para calcular los componentes principales es:
- **Paso 1: Estandarizar los datos.**

Cálculo de los Componentes Principales

- El proceso para calcular los componentes principales es:
- **Paso 1: Estandarizar los datos.**
 - Es crucial que cada característica tenga media cero y desviación estándar uno.

Cálculo de los Componentes Principales

- El proceso para calcular los componentes principales es:
- **Paso 1: Estandarizar los datos.**
 - Es crucial que cada característica tenga media cero y desviación estándar uno.
- **Paso 2: Calcular la matriz de covarianza.**

Cálculo de los Componentes Principales

- El proceso para calcular los componentes principales es:
- **Paso 1: Estandarizar los datos.**
 - Es crucial que cada característica tenga media cero y desviación estándar uno.
- **Paso 2: Calcular la matriz de covarianza.**
 - Esta matriz $p \times p$ describe las relaciones lineales entre las características.

Cálculo de los Componentes Principales

- El proceso para calcular los componentes principales es:
- **Paso 1: Estandarizar los datos.**
 - Es crucial que cada característica tenga media cero y desviación estándar uno.
- **Paso 2: Calcular la matriz de covarianza.**
 - Esta matriz $p \times p$ describe las relaciones lineales entre las características.
- **Paso 3: Realizar la descomposición espectral (Eigen-decomposition).**

Cálculo de los Componentes Principales

- El proceso para calcular los componentes principales es:
- **Paso 1: Estandarizar los datos.**
 - Es crucial que cada característica tenga media cero y desviación estándar uno.
- **Paso 2: Calcular la matriz de covarianza.**
 - Esta matriz $p \times p$ describe las relaciones lineales entre las características.
- **Paso 3: Realizar la descomposición espectral (Eigen-decomposition).**
 - Los **vectores propios (eigenvectors)** de la matriz de covarianza son los vectores de loadings (ϕ_m), es decir, las direcciones de los componentes principales.

Cálculo de los Componentes Principales

- El proceso para calcular los componentes principales es:
- **Paso 1: Estandarizar los datos.**
 - Es crucial que cada característica tenga media cero y desviación estándar uno.
- **Paso 2: Calcular la matriz de covarianza.**
 - Esta matriz $p \times p$ describe las relaciones lineales entre las características.
- **Paso 3: Realizar la descomposición espectral (Eigen-decomposition).**
 - Los **vectores propios (eigenvectors)** de la matriz de covarianza son los vectores de loadings (ϕ_m), es decir, las direcciones de los componentes principales.
 - Los **valores propios (eigenvalues)** indican la cantidad de varianza capturada por cada componente principal.

Formalización del Problema de Optimización en PCA

Formalización del Problema de Optimización en PCA

- **Objetivo:** Maximizar la varianza de los datos proyectados.

Formalización del Problema de Optimización en PCA

- **Objetivo:** Maximizar la varianza de los datos proyectados.
- Para el primer componente principal, la varianza de los scores ($z_{i1} = \phi_1^T \mathbf{x}_i$) es:

$$\text{Var}(Z_1) = \frac{1}{n} \sum_{i=1}^n (\phi_1^T \mathbf{x}_i)^2 = \phi_1^T \left(\frac{1}{n} \mathbf{X}^T \mathbf{X} \right) \phi_1$$

Formalización del Problema de Optimización en PCA

- **Objetivo:** Maximizar la varianza de los datos proyectados.
- Para el primer componente principal, la varianza de los scores ($z_{i1} = \phi_1^T \mathbf{x}_i$) es:

$$\text{Var}(Z_1) = \frac{1}{n} \sum_{i=1}^n (\phi_1^T \mathbf{x}_i)^2 = \phi_1^T \left(\frac{1}{n} \mathbf{X}^T \mathbf{X} \right) \phi_1$$

- Notamos que $\mathbf{S} = \frac{1}{n} \mathbf{X}^T \mathbf{X}$ es la matriz de covarianza muestral (para datos centrados).

Formalización del Problema de Optimización en PCA

- **Objetivo:** Maximizar la varianza de los datos proyectados.
- Para el primer componente principal, la varianza de los scores ($z_{i1} = \phi_1^T \mathbf{x}_i$) es:

$$\text{Var}(Z_1) = \frac{1}{n} \sum_{i=1}^n (\phi_1^T \mathbf{x}_i)^2 = \phi_1^T \left(\frac{1}{n} \mathbf{X}^T \mathbf{X} \right) \phi_1$$

- Notamos que $\mathbf{S} = \frac{1}{n} \mathbf{X}^T \mathbf{X}$ es la matriz de covarianza muestral (para datos centrados).
- El problema de optimización para el primer componente es:

$$\underset{\phi_1}{\text{maximizar}} \quad \left\{ \phi_1^T \mathbf{S} \phi_1 \right\}$$

Formalización del Problema de Optimización en PCA

- **Objetivo:** Maximizar la varianza de los datos proyectados.
- Para el primer componente principal, la varianza de los scores ($z_{i1} = \phi_1^T \mathbf{x}_i$) es:

$$\text{Var}(Z_1) = \frac{1}{n} \sum_{i=1}^n (\phi_1^T \mathbf{x}_i)^2 = \phi_1^T \left(\frac{1}{n} \mathbf{X}^T \mathbf{X} \right) \phi_1$$

- Notamos que $\mathbf{S} = \frac{1}{n} \mathbf{X}^T \mathbf{X}$ es la matriz de covarianza muestral (para datos centrados).
- El problema de optimización para el primer componente es:

$$\underset{\phi_1}{\text{maximizar}} \quad \left\{ \phi_1^T \mathbf{S} \phi_1 \right\}$$

- Sujeto a la restricción de normalización para evitar soluciones infinitas:

$$\phi_1^T \phi_1 = \sum_{j=1}^p \phi_{j1}^2 = 1$$

La Solución: El Teorema Espectral

- El problema de optimización restringida se resuelve usando **multiplicadores de Lagrange**.

La Solución: El Teorema Espectral

- El problema de optimización restringida se resuelve usando **multiplicadores de Lagrange**.
- Construimos la función Lagrangiana \mathcal{L} :

$$\mathcal{L}(\phi_1, \lambda) = \phi_1^T \mathbf{S} \phi_1 - \lambda(\phi_1^T \phi_1 - 1)$$

La Solución: El Teorema Espectral

- El problema de optimización restringida se resuelve usando **multiplicadores de Lagrange**.
- Construimos la función Lagrangiana \mathcal{L} :

$$\mathcal{L}(\phi_1, \lambda) = \phi_1^T \mathbf{S} \phi_1 - \lambda(\phi_1^T \phi_1 - 1)$$

- Derivando con respecto a ϕ_1 e igualando a cero para encontrar el máximo:

$$\nabla_{\phi_1} \mathcal{L} = 2\mathbf{S}\phi_1 - 2\lambda\phi_1 = 0 \quad \implies \quad \mathbf{S}\phi_1 = \lambda\phi_1$$

La Solución: El Teorema Espectral

- El problema de optimización restringida se resuelve usando **multiplicadores de Lagrange**.
- Construimos la función Lagrangiana \mathcal{L} :

$$\mathcal{L}(\phi_1, \lambda) = \phi_1^T \mathbf{S} \phi_1 - \lambda(\phi_1^T \phi_1 - 1)$$

- Derivando con respecto a ϕ_1 e igualando a cero para encontrar el máximo:

$$\nabla_{\phi_1} \mathcal{L} = 2\mathbf{S}\phi_1 - 2\lambda\phi_1 = 0 \quad \implies \quad \mathbf{S}\phi_1 = \lambda\phi_1$$

- **¡Este es el resultado fundamental!** La ecuación anterior es la definición de un problema de **valores y vectores propios**.

La Solución: El Teorema Espectral

- El problema de optimización restringida se resuelve usando **multiplicadores de Lagrange**.
- Construimos la función Lagrangiana \mathcal{L} :

$$\mathcal{L}(\phi_1, \lambda) = \phi_1^T \mathbf{S} \phi_1 - \lambda(\phi_1^T \phi_1 - 1)$$

- Derivando con respecto a ϕ_1 e igualando a cero para encontrar el máximo:

$$\nabla_{\phi_1} \mathcal{L} = 2\mathbf{S}\phi_1 - 2\lambda\phi_1 = 0 \quad \implies \quad \mathbf{S}\phi_1 = \lambda\phi_1$$

- **¡Este es el resultado fundamental!** La ecuación anterior es la definición de un problema de **valores y vectores propios**.
- El vector de loadings ϕ_1 debe ser un **vector propio** (eigenvector) de la matriz de covarianza \mathbf{S} .

La Solución: El Teorema Espectral

- El problema de optimización restringida se resuelve usando **multiplicadores de Lagrange**.
- Construimos la función Lagrangiana \mathcal{L} :

$$\mathcal{L}(\phi_1, \lambda) = \phi_1^T \mathbf{S} \phi_1 - \lambda(\phi_1^T \phi_1 - 1)$$

- Derivando con respecto a ϕ_1 e igualando a cero para encontrar el máximo:

$$\nabla_{\phi_1} \mathcal{L} = 2\mathbf{S}\phi_1 - 2\lambda\phi_1 = 0 \quad \implies \quad \mathbf{S}\phi_1 = \lambda\phi_1$$

- **¡Este es el resultado fundamental!** La ecuación anterior es la definición de un problema de **valores y vectores propios**.
- El vector de loadings ϕ_1 debe ser un **vector propio** (eigenvector) de la matriz de covarianza \mathbf{S} .
- Para maximizar la varianza ($\phi_1^T \mathbf{S} \phi_1 = \lambda$), debemos elegir el eigenvector correspondiente al **valor propio (eigenvalue) más**

Pausa de 15 minutos

Varianza Explicada (PVE)

Varianza Explicada (PVE)

- ¿Cuánta información perdemos al proyectar los datos a un subespacio de menor dimensión?

Varianza Explicada (PVE)

- ¿Cuánta información perdemos al proyectar los datos a un subespacio de menor dimensión?
- La **Proporción de Varianza Explicada (PVE)** por el m -ésimo componente principal es la fracción de la varianza total que captura ese componente.

$$\text{PVE}_m = \frac{\lambda_m}{\sum_{j=1}^p \lambda_j} = \frac{\text{Varianza del CP } m}{\text{Varianza Total}}$$

Varianza Explicada (PVE)

- ¿Cuánta información perdemos al proyectar los datos a un subespacio de menor dimensión?
- La **Proporción de Varianza Explicada (PVE)** por el m -ésimo componente principal es la fracción de la varianza total que captura ese componente.

$$\text{PVE}_m = \frac{\lambda_m}{\sum_{j=1}^p \lambda_j} = \frac{\text{Varianza del CP } m}{\text{Varianza Total}}$$

- La PVE acumulada nos ayuda a decidir cuántos componentes conservar. Por ejemplo, podríamos querer retener suficientes componentes para explicar el 80-95% de la varianza total.

Selección de Componentes: El Scree Plot

Selección de Componentes: El Scree Plot

- Un **Scree Plot** es un gráfico que muestra la varianza explicada por cada componente principal, ordenados de mayor a menor.

Selección de Componentes: El Scree Plot

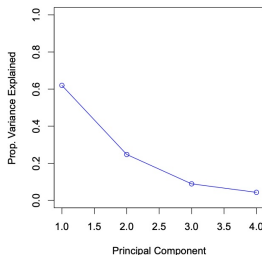
- Un **Scree Plot** es un gráfico que muestra la varianza explicada por cada componente principal, ordenados de mayor a menor.
- Es la herramienta visual principal para decidir cuántos componentes retener.

Selección de Componentes: El Scree Plot

- Un **Scree Plot** es un gráfico que muestra la varianza explicada por cada componente principal, ordenados de mayor a menor.
- Es la herramienta visual principal para decidir cuántos componentes retener.
- Buscamos un “codo” (elbow) en el gráfico: el punto donde la PVE marginal de componentes adicionales disminuye drásticamente.

Selección de Componentes: El Scree Plot

- Un **Scree Plot** es un gráfico que muestra la varianza explicada por cada componente principal, ordenados de mayor a menor.
- Es la herramienta visual principal para decidir cuántos componentes retener.
- Buscamos un “codo” (elbow) en el gráfico: el punto donde la PVE marginal de componentes adicionales disminuye drásticamente.
- Los componentes a la izquierda del codo son los que retienen la mayor parte de la “señal”, mientras que los de la derecha podrían ser principalmente “ruido”.



Consideración Práctica: Escalar las Variables

Consideración Práctica: Escalar las Variables

- PCA es **muy sensible a la escala** de las variables.

Consideración Práctica: Escalar las Variables

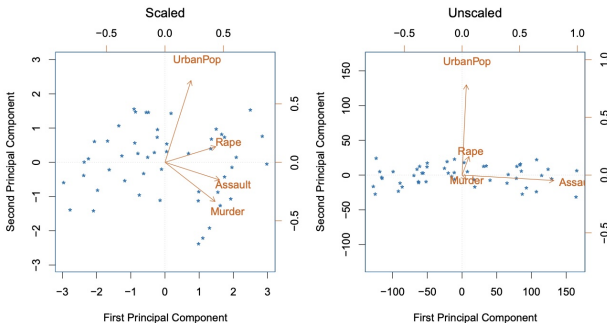
- PCA es **muy sensible a la escala** de las variables.
- Si las variables no se escalan, aquellas con mayor varianza dominarán el primer componente principal, independientemente de su importancia real.

Consideración Práctica: Escalar las Variables

- PCA es **muy sensible a la escala** de las variables.
- Si las variables no se escalan, aquellas con mayor varianza dominarán el primer componente principal, independientemente de su importancia real.
- **Ejemplo:** Si una variable está en kilogramos y otra en metros, la elección de unidades (gramos vs. kilómetros) cambiaría drásticamente el resultado de PCA.

Consideración Práctica: Escalar las Variables

- PCA es **muy sensible a la escala** de las variables.
- Si las variables no se escalan, aquellas con mayor varianza dominarán el primer componente principal, independientemente de su importancia real.
- **Ejemplo:** Si una variable está en kilogramos y otra en metros, la elección de unidades (gramos vs. kilómetros) cambiaría drásticamente el resultado de PCA.
- **Regla general:** Siempre estandarizar las variables (media 0, desviación estándar 1) antes de aplicar PCA, a menos que todas las variables estén en las mismas unidades y se desee que la varianza original influya.



Limitaciones de PCA y Métodos No Lineales

Limitaciones de PCA y Métodos No Lineales

- PCA es una técnica lineal, por lo que asume que la estructura subyacente de los datos puede ser bien representada por un subespacio lineal.

Limitaciones de PCA y Métodos No Lineales

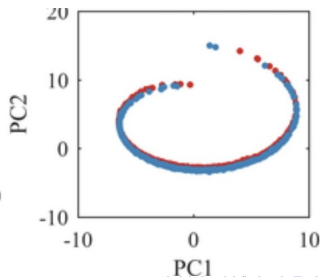
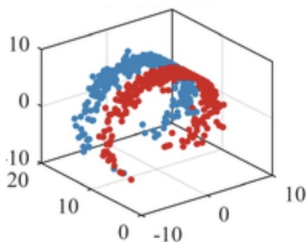
- PCA es una técnica lineal, por lo que asume que la estructura subyacente de los datos puede ser bien representada por un subespacio lineal.
- ¿Qué pasa si los datos se encuentran en una **variedad (manifold)** no lineal y curva?

Limitaciones de PCA y Métodos No Lineales

- PCA es una técnica lineal, por lo que asume que la estructura subyacente de los datos puede ser bien representada por un subespacio lineal.
- ¿Qué pasa si los datos se encuentran en una **variedad (manifold)** no lineal y curva?
- PCA “aplastará” la estructura, perdiendo información importante sobre la proximidad de los puntos.

Limitaciones de PCA y Métodos No Lineales

- PCA es una técnica lineal, por lo que asume que la estructura subyacente de los datos puede ser bien representada por un subespacio lineal.
- ¿Qué pasa si los datos se encuentran en una **variedad (manifold)** no lineal y curva?
- PCA “aplastará” la estructura, perdiendo información importante sobre la proximidad de los puntos.
- Para estos casos, necesitamos técnicas de reducción de dimensionalidad no lineales.



t-SNE: Visualización de Datos de Alta Dimensión

t-SNE: Visualización de Datos de Alta Dimensión

- **t-SNE (t-distributed Stochastic Neighbor Embedding)** es una técnica no lineal, principalmente para visualización en 2D o 3D.

t-SNE: Visualización de Datos de Alta Dimensión

- **t-SNE (t-distributed Stochastic Neighbor Embedding)** es una técnica no lineal, principalmente para visualización en 2D o 3D.
- **Intuición:** Intenta preservar las “vecindades” de los puntos. Si dos puntos son cercanos en el espacio de alta dimensión, t-SNE intentará que sigan siendo cercanos en el espacio de baja dimensión.

t-SNE: Visualización de Datos de Alta Dimensión

- **t-SNE (t-distributed Stochastic Neighbor Embedding)** es una técnica no lineal, principalmente para visualización en 2D o 3D.
- **Intuición:** Intenta preservar las “vecindades” de los puntos. Si dos puntos son cercanos en el espacio de alta dimensión, t-SNE intentará que sigan siendo cercanos en el espacio de baja dimensión.
- Funciona bien para revelar la estructura de **clusters locales** en los datos.

t-SNE: Visualización de Datos de Alta Dimensión

- **t-SNE (t-distributed Stochastic Neighbor Embedding)** es una técnica no lineal, principalmente para visualización en 2D o 3D.
- **Intuición:** Intenta preservar las “vecindades” de los puntos. Si dos puntos son cercanos en el espacio de alta dimensión, t-SNE intentará que sigan siendo cercanos en el espacio de baja dimensión.
- Funciona bien para revelar la estructura de **clusters locales** en los datos.
- **Consideraciones:**

t-SNE: Visualización de Datos de Alta Dimensión

- **t-SNE (t-distributed Stochastic Neighbor Embedding)** es una técnica no lineal, principalmente para visualización en 2D o 3D.
- **Intuición:** Intenta preservar las “vecindades” de los puntos. Si dos puntos son cercanos en el espacio de alta dimensión, t-SNE intentará que sigan siendo cercanos en el espacio de baja dimensión.
- Funciona bien para revelar la estructura de **clusters locales** en los datos.
- **Consideraciones:**
 - Es computacionalmente intensivo.

t-SNE: Visualización de Datos de Alta Dimensión

- **t-SNE (t-distributed Stochastic Neighbor Embedding)** es una técnica no lineal, principalmente para visualización en 2D o 3D.
- **Intuición:** Intenta preservar las “vecindades” de los puntos. Si dos puntos son cercanos en el espacio de alta dimensión, t-SNE intentará que sigan siendo cercanos en el espacio de baja dimensión.
- Funciona bien para revelar la estructura de **clusters locales** en los datos.
- **Consideraciones:**
 - Es computacionalmente intensivo.
 - Los tamaños de los clusters y las distancias entre ellos en el gráfico final no siempre tienen un significado global directo. ¡La interpretación requiere cautela!

t-SNE: Formalización Matemática

t-SNE: Formalización Matemática

- **1. Similitud en alta dimensión (p_{ij}):** Se modela la probabilidad de que \mathbf{x}_j sea vecino de \mathbf{x}_i con una distribución Gaussiana. Tras simetrizar:

$$p_{ij} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_k - \mathbf{x}_i\|^2/2\sigma^2)}$$

- **1. Similitud en alta dimensión (p_{ij}):** Se modela la probabilidad de que \mathbf{x}_j sea vecino de \mathbf{x}_i con una distribución Gaussiana. Tras simetrizar:

$$p_{ij} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2)}{\sum_{k \neq l} \exp(-\|\mathbf{x}_k - \mathbf{x}_l\|^2 / 2\sigma^2)}$$

- **2. Similitud en baja dimensión (q_{ij}):** En el espacio 2D/3D, se usa una distribución t-Student (cola pesada) para permitir que puntos que eran moderadamente distantes se alejen más, ayudando a separar clusters.

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}}$$

- **1. Similitud en alta dimensión (p_{ij}):** Se modela la probabilidad de que \mathbf{x}_j sea vecino de \mathbf{x}_i con una distribución Gaussiana. Tras simetrizar:

$$p_{ij} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_k - \mathbf{x}_i\|^2 / 2\sigma^2)}$$

- **2. Similitud en baja dimensión (q_{ij}):** En el espacio 2D/3D, se usa una distribución t-Student (cola pesada) para permitir que puntos que eran moderadamente distantes se alejen más, ayudando a separar clusters.

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq i} (1 + \|\mathbf{y}_k - \mathbf{y}_i\|^2)^{-1}}$$

- **3. Optimización:** Se minimiza la divergencia de Kullback-Leibler (KL) entre las distribuciones de probabilidad P y Q , que mide la “diferencia” entre ellas.

$$C = \text{KL}(P \| Q) = \sum p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

UMAP: Una Alternativa Moderna a t-SNE

UMAP: Una Alternativa Moderna a t-SNE

- **UMAP (Uniform Manifold Approximation and Projection)** es otra técnica no lineal de reducción de dimensionalidad.

UMAP: Una Alternativa Moderna a t-SNE

- **UMAP (Uniform Manifold Approximation and Projection)** es otra técnica no lineal de reducción de dimensionalidad.
- Comparte objetivos similares con t-SNE, pero se basa en una teoría matemática diferente (topología de variedades).

UMAP: Una Alternativa Moderna a t-SNE

- **UMAP (Uniform Manifold Approximation and Projection)** es otra técnica no lineal de reducción de dimensionalidad.
- Comparte objetivos similares con t-SNE, pero se basa en una teoría matemática diferente (topología de variedades).
- **Ventajas sobre t-SNE:**

UMAP: Una Alternativa Moderna a t-SNE

- **UMAP (Uniform Manifold Approximation and Projection)** es otra técnica no lineal de reducción de dimensionalidad.
- Comparte objetivos similares con t-SNE, pero se basa en una teoría matemática diferente (topología de variedades).
- **Ventajas sobre t-SNE:**
 - **Velocidad:** Generalmente mucho más rápido, lo que permite su uso en datasets más grandes.

UMAP: Una Alternativa Moderna a t-SNE

- **UMAP (Uniform Manifold Approximation and Projection)** es otra técnica no lineal de reducción de dimensionalidad.
- Comparte objetivos similares con t-SNE, pero se basa en una teoría matemática diferente (topología de variedades).
- **Ventajas sobre t-SNE:**
 - **Velocidad:** Generalmente mucho más rápido, lo que permite su uso en datasets más grandes.
 - **Preservación de la estructura global:** UMAP a menudo hace un mejor trabajo preservando la estructura a gran escala de los datos, no solo las vecindades locales.

UMAP: Una Alternativa Moderna a t-SNE

- **UMAP (Uniform Manifold Approximation and Projection)** es otra técnica no lineal de reducción de dimensionalidad.
- Comparte objetivos similares con t-SNE, pero se basa en una teoría matemática diferente (topología de variedades).
- **Ventajas sobre t-SNE:**
 - **Velocidad:** Generalmente mucho más rápido, lo que permite su uso en datasets más grandes.
 - **Preservación de la estructura global:** UMAP a menudo hace un mejor trabajo preservando la estructura a gran escala de los datos, no solo las vecindades locales.
 - Es más versátil y puede usarse para más que solo visualización (ej. como paso de preprocesamiento).

UMAP: Una Alternativa Moderna a t-SNE

- **UMAP (Uniform Manifold Approximation and Projection)** es otra técnica no lineal de reducción de dimensionalidad.
- Comparte objetivos similares con t-SNE, pero se basa en una teoría matemática diferente (topología de variedades).
- **Ventajas sobre t-SNE:**
 - **Velocidad:** Generalmente mucho más rápido, lo que permite su uso en datasets más grandes.
 - **Preservación de la estructura global:** UMAP a menudo hace un mejor trabajo preservando la estructura a gran escala de los datos, no solo las vecindades locales.
 - Es más versátil y puede usarse para más que solo visualización (ej. como paso de preprocesamiento).
- Se está convirtiendo en la herramienta de elección para muchas tareas de visualización.

UMAP: Formalización Matemática

UMAP: Formalización Matemática

- **1. Grafo en alta dimensión (v_{ij}):** UMAP construye un grafo de vecinos y define una similitud “difusa” que se desvanece con la distancia. Tras simetrizar:

$$v_{ij} = v_{j|i} + v_{i|j} - v_{j|i}v_{i|j} \quad \text{donde} \quad v_{j|i} \approx \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|}{\sigma_i}\right)$$

UMAP: Formalización Matemática

- **1. Grafo en alta dimensión (v_{ij}):** UMAP construye un grafo de vecinos y define una similitud “difusa” que se desvanece con la distancia. Tras simetrizar:

$$v_{ij} = v_{j|i} + v_{i|j} - v_{j|i}v_{i|j} \quad \text{donde} \quad v_{j|i} \approx \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|}{\sigma_i}\right)$$

- **2. Grafo en baja dimensión (w_{ij}):** Define una estructura similar en el espacio de baja dimensión, usando una familia de curvas parametrizada:

$$w_{ij} = \left(1 + a\|\mathbf{y}_i - \mathbf{y}_j\|^{2b}\right)^{-1}$$

UMAP: Formalización Matemática

- **1. Grafo en alta dimensión (v_{ij}):** UMAP construye un grafo de vecinos y define una similitud “difusa” que se desvanece con la distancia. Tras simetrizar:

$$v_{ij} = v_{j|i} + v_{i|j} - v_{j|i}v_{i|j} \quad \text{donde} \quad v_{j|i} \approx \exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|}{\sigma_i}\right)$$

- **2. Grafo en baja dimensión (w_{ij}):** Define una estructura similar en el espacio de baja dimensión, usando una familia de curvas parametrizada:

$$w_{ij} = \left(1 + a\|\mathbf{y}_i - \mathbf{y}_j\|^{2b}\right)^{-1}$$

- **3. Optimización:** UMAP minimiza la **Cross-Entropy (CE)** binaria entre las dos representaciones. Esto es clave para su eficiencia y mejor preservación de la estructura global.

$$C(V, W) = \sum_{i < j} \left[v_{ij} \log\left(\frac{v_{ij}}{w_{ij}}\right) + (1 - v_{ij}) \log\left(\frac{1 - v_{ij}}{1 - w_{ij}}\right) \right]$$

Comparación Visual: PCA vs. t-SNE vs. UMAP

Comparación Visual: PCA vs. t-SNE vs. UMAP

- Para un mismo dataset de alta dimensionalidad (como MNIST, dígitos escritos a mano), cada técnica revela una estructura diferente.

Comparación Visual: PCA vs. t-SNE vs. UMAP

- Para un mismo dataset de alta dimensionalidad (como MNIST, dígitos escritos a mano), cada técnica revela una estructura diferente.
- **PCA:** Puede mostrar una separación general, pero los grupos a menudo se superponen.

Comparación Visual: PCA vs. t-SNE vs. UMAP

- Para un mismo dataset de alta dimensionalidad (como MNIST, dígitos escritos a mano), cada técnica revela una estructura diferente.
- **PCA:** Puede mostrar una separación general, pero los grupos a menudo se superponen.
- **t-SNE:** Muestra clusters locales muy definidos y separados.

Comparación Visual: PCA vs. t-SNE vs. UMAP

- Para un mismo dataset de alta dimensionalidad (como MNIST, dígitos escritos a mano), cada técnica revela una estructura diferente.
- **PCA:** Puede mostrar una separación general, pero los grupos a menudo se superponen.
- **t-SNE:** Muestra clusters locales muy definidos y separados.
- **UMAP:** También muestra clusters definidos, pero a menudo con una mejor representación de cómo se relacionan los clusters entre sí.

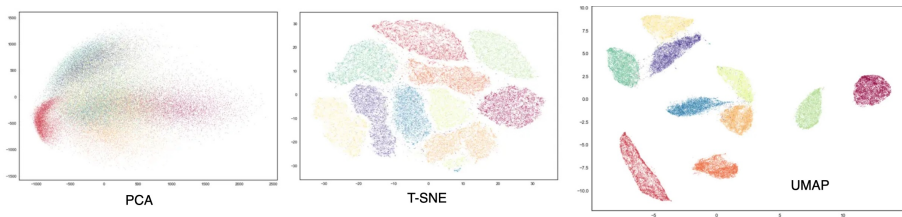


Figure: Visualización del dataset MNIST usando PCA, t-SNE y UMAP.

Aplicaciones Prácticas y Resumen

- **Preprocesamiento para Supervisado:** Reducir características con PCA antes de entrenar un clasificador o regresor puede mejorar el rendimiento y reducir el tiempo de entrenamiento.

- **Preprocesamiento para Supervisado:** Reducir características con PCA antes de entrenar un clasificador o regresor puede mejorar el rendimiento y reducir el tiempo de entrenamiento.
- **Visualización de Datos:**

- **Preprocesamiento para Supervisado:** Reducir características con PCA antes de entrenar un clasificador o regresor puede mejorar el rendimiento y reducir el tiempo de entrenamiento.
- **Visualización de Datos:**
 - Usar PCA para una primera exploración rápida y lineal.

- **Preprocesamiento para Supervisado:** Reducir características con PCA antes de entrenar un clasificador o regresor puede mejorar el rendimiento y reducir el tiempo de entrenamiento.
- **Visualización de Datos:**
 - Usar PCA para una primera exploración rápida y lineal.
 - Usar t-SNE o UMAP para descubrir e investigar estructuras de clusters complejas en 2D/3D (ej. datos genómicos, embeddings de texto).

- **Preprocesamiento para Supervisado:** Reducir características con PCA antes de entrenar un clasificador o regresor puede mejorar el rendimiento y reducir el tiempo de entrenamiento.
- **Visualización de Datos:**
 - Usar PCA para una primera exploración rápida y lineal.
 - Usar t-SNE o UMAP para descubrir e investigar estructuras de clusters complejas en 2D/3D (ej. datos genómicos, embeddings de texto).
- **Compresión de Datos:**

- **Preprocesamiento para Supervisado:** Reducir características con PCA antes de entrenar un clasificador o regresor puede mejorar el rendimiento y reducir el tiempo de entrenamiento.
- **Visualización de Datos:**
 - Usar PCA para una primera exploración rápida y lineal.
 - Usar t-SNE o UMAP para descubrir e investigar estructuras de clusters complejas en 2D/3D (ej. datos genómicos, embeddings de texto).
- **Compresión de Datos:**
 - En procesamiento de imágenes, PCA (conocido como Eigenfaces) puede usarse para comprimir y reconocer rostros.

- **Preprocesamiento para Supervisado:** Reducir características con PCA antes de entrenar un clasificador o regresor puede mejorar el rendimiento y reducir el tiempo de entrenamiento.
- **Visualización de Datos:**
 - Usar PCA para una primera exploración rápida y lineal.
 - Usar t-SNE o UMAP para descubrir e investigar estructuras de clusters complejas en 2D/3D (ej. datos genómicos, embeddings de texto).
- **Compresión de Datos:**
 - En procesamiento de imágenes, PCA (conocido como Eigenfaces) puede usarse para comprimir y reconocer rostros.
- **Resumen:** Elegir la técnica adecuada (lineal vs. no lineal) depende del objetivo y la naturaleza de los datos.

Bonus Track: Un Vistazo a Otros Métodos No Supervisados

Las siguientes diapositivas presentan brevemente otros métodos importantes que no podremos cubrir en detalle.

El objetivo es que conozcan sus **nombres**, **aplicaciones clave** y una **imagen conceptual**.

1. Análisis de Componentes Independientes (ICA)

1. Análisis de Componentes Independientes (ICA)

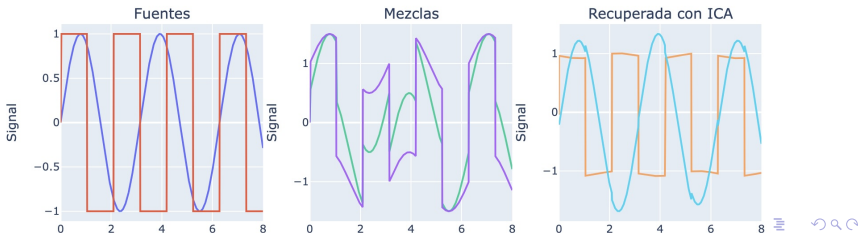
- **¿Qué es?** Una técnica para separar una señal multivariada en subcomponentes aditivos que son estadísticamente **independientes**.

1. Análisis de Componentes Independientes (ICA)

- **¿Qué es?** Una técnica para separar una señal multivariada en subcomponentes aditivos que son estadísticamente **independientes**.
- **Diferencia con PCA:** PCA maximiza la varianza y produce componentes *no correlacionados*. ICA maximiza la independencia estadística y puede encontrar componentes aunque no sean ortogonales.

1. Análisis de Componentes Independientes (ICA)

- **¿Qué es?** Una técnica para separar una señal multivariada en subcomponentes aditivos que son estadísticamente **independientes**.
- **Diferencia con PCA:** PCA maximiza la varianza y produce componentes *no correlacionados*. ICA maximiza la independencia estadística y puede encontrar componentes aunque no sean ortogonales.
- **Aplicación Clásica:** El “problema de la fiesta” (cocktail party problem). ICA puede tomar grabaciones de varios micrófonos en una sala y separar las voces de las distintas personas en pistas de audio independientes.



2. Modelos de Mezcla Gaussiana (GMM)

2. Modelos de Mezcla Gaussiana (GMM)

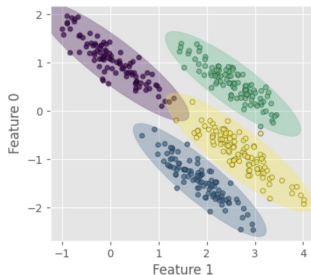
- **¿Qué es?** Un método de clustering **probabilístico** o “suave” (soft clustering). Modela los datos como generados por una mezcla de múltiples distribuciones Gaussianas.

2. Modelos de Mezcla Gaussiana (GMM)

- **¿Qué es?** Un método de clustering **probabilístico** o “suave” (soft clustering). Modela los datos como generados por una mezcla de múltiples distribuciones Gaussianas.
- **¿Cómo funciona?** Cada punto tiene una probabilidad de pertenecer a cada uno de los clusters (Gaussianas). Se ajusta con el algoritmo **Expectation-Maximization (EM)**.

2. Modelos de Mezcla Gaussiana (GMM)

- **¿Qué es?** Un método de clustering **probabilístico** o “suave” (soft clustering). Modela los datos como generados por una mezcla de múltiples distribuciones Gaussianas.
- **¿Cómo funciona?** Cada punto tiene una probabilidad de pertenecer a cada uno de los clusters (Gaussianas). Se ajusta con el algoritmo **Expectation-Maximization (EM)**.
- **Ventaja sobre K-Means:** Es más flexible. Mientras que K-Means asume clusters esféricos, GMM puede modelar clusters con formas **elípticas** y diferentes orientaciones.



3. Autoencoders para Reducción de Dimensión

3. Autoencoders para Reducción de Dimensión

- **¿Qué es?** Un tipo de red neuronal usada para aprender representaciones comprimidas de los datos de forma no supervisada.

3. Autoencoders para Reducción de Dimensión

- **¿Qué es?** Un tipo de red neuronal usada para aprender representaciones comprimidas de los datos de forma no supervisada.
- **Arquitectura:** Un **codificador (encoder)** comprime el input a un “cuello de botella” (bottleneck) de baja dimensión. Un **decodificador (decoder)** intenta reconstruir el input original desde esa representación comprimida.

3. Autoencoders para Reducción de Dimensión

- **¿Qué es?** Un tipo de red neuronal usada para aprender representaciones comprimidas de los datos de forma no supervisada.
- **Arquitectura:** Un **codificador (encoder)** comprime el input a un “cuello de botella” (bottleneck) de baja dimensión. Un **decodificador (decoder)** intenta reconstruir el input original desde esa representación comprimida.
- **Poder:** Aprende mapeos **no lineales**, lo que lo hace mucho más potente que PCA para datos complejos como imágenes o texto.

4. Aprendizaje de Características y Clustering Profundo

4. Aprendizaje de Características y Clustering Profundo

- **¿Qué es?** Un enfoque híbrido que une Redes Neuronales Profundas (Deep Learning) con algoritmos de clustering.

4. Aprendizaje de Características y Clustering Profundo

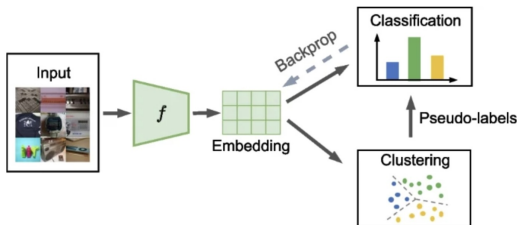
- **¿Qué es?** Un enfoque híbrido que une Redes Neuronales Profundas (Deep Learning) con algoritmos de clustering.
- **Idea Central:** Entrenar una red neuronal (ej. un autoencoder) para que, de forma simultánea, aprenda una representación de baja dimensión y optimice una función de pérdida de clustering.

4. Aprendizaje de Características y Clustering Profundo

- **¿Qué es?** Un enfoque híbrido que une Redes Neuronales Profundas (Deep Learning) con algoritmos de clustering.
- **Idea Central:** Entrenar una red neuronal (ej. un autoencoder) para que, de forma simultánea, aprenda una representación de baja dimensión y optimice una función de pérdida de clustering.
- **Objetivo:** La red aprende características que son inherentemente “amigables para el clustering”, es decir, que forman grupos compactos y bien separados en el espacio latente.

4. Aprendizaje de Características y Clustering Profundo

- **¿Qué es?** Un enfoque híbrido que une Redes Neuronales Profundas (Deep Learning) con algoritmos de clustering.
- **Idea Central:** Entrenar una red neuronal (ej. un autoencoder) para que, de forma simultánea, aprenda una representación de baja dimensión y optimice una función de pérdida de clustering.
- **Objetivo:** La red aprende características que son inherentemente “amigables para el clustering”, es decir, que forman grupos compactos y bien separados en el espacio latente.
- **Aplicación:** Muy útil para agrupar datos complejos y no estructurados (imágenes, documentos) donde las características relevantes no son obvias.



Resumen de Aplicaciones Avanzadas

- **Análisis de Componentes Independientes (ICA)**

- **Análisis de Componentes Independientes (ICA)**
 - Separación de señales de audio.

- **Análisis de Componentes Independientes (ICA)**

- Separación de señales de audio.
- Análisis de señales biomédicas (EEG, fMRI).

- **Análisis de Componentes Independientes (ICA)**
 - Separación de señales de audio.
 - Análisis de señales biomédicas (EEG, fMRI).
- **Modelos de Mezcla Gaussiana (GMM)**

- **Análisis de Componentes Independientes (ICA)**
 - Separación de señales de audio.
 - Análisis de señales biomédicas (EEG, fMRI).
- **Modelos de Mezcla Gaussiana (GMM)**
 - Segmentación de imágenes por color/textura.

- **Análisis de Componentes Independientes (ICA)**

- Separación de señales de audio.
- Análisis de señales biomédicas (EEG, fMRI).

- **Modelos de Mezcla Gaussiana (GMM)**

- Segmentación de imágenes por color/textura.
- Modelado de perfiles de riesgo o de clientes.

- **Análisis de Componentes Independientes (ICA)**
 - Separación de señales de audio.
 - Análisis de señales biomédicas (EEG, fMRI).
- **Modelos de Mezcla Gaussiana (GMM)**
 - Segmentación de imágenes por color/textura.
 - Modelado de perfiles de riesgo o de clientes.
- **Autoencoders y Clustering Profundo**

- **Análisis de Componentes Independientes (ICA)**
 - Separación de señales de audio.
 - Análisis de señales biomédicas (EEG, fMRI).
- **Modelos de Mezcla Gaussiana (GMM)**
 - Segmentación de imágenes por color/textura.
 - Modelado de perfiles de riesgo o de clientes.
- **Autoencoders y Clustering Profundo**
 - Detección de anomalías en ciberseguridad o finanzas.

- **Análisis de Componentes Independientes (ICA)**
 - Separación de señales de audio.
 - Análisis de señales biomédicas (EEG, fMRI).
- **Modelos de Mezcla Gaussiana (GMM)**
 - Segmentación de imágenes por color/textura.
 - Modelado de perfiles de riesgo o de clientes.
- **Autoencoders y Clustering Profundo**
 - Detección de anomalías en ciberseguridad o finanzas.
 - Búsqueda de imágenes por similitud visual.

- **Análisis de Componentes Independientes (ICA)**
 - Separación de señales de audio.
 - Análisis de señales biomédicas (EEG, fMRI).
- **Modelos de Mezcla Gaussiana (GMM)**
 - Segmentación de imágenes por color/textura.
 - Modelado de perfiles de riesgo o de clientes.
- **Autoencoders y Clustering Profundo**
 - Detección de anomalías en ciberseguridad o finanzas.
 - Búsqueda de imágenes por similitud visual.
 - Generación de datos sintéticos.