UNIVERSIDAD DEL VALLE DE GUATEMALA

CC3074 – Minería de Datos

Sección 10

MSc. Linette García Pérez

Otros Algoritmos de Aprendizaje

Daniela Ramírez de León

23053

GUATEMALA, 13 de febrero de 2026

Introducción

El aprendizaje no supervisado constituye una de las áreas fundamentales dentro de la minería de datos y el aprendizaje automático, ya que permite identificar patrones, estructuras y relaciones ocultas en conjuntos de datos sin necesidad de etiquetas previamente definidas. A diferencia del aprendizaje supervisado, donde se dispone de una variable objetivo, los métodos no supervisados buscan descubrir información inherente en los datos mediante técnicas de agrupamiento, reducción de dimensionalidad y separación de fuentes latentes.

En el análisis moderno de datos, la reducción de dimensionalidad desempeña un papel crucial, especialmente cuando se trabaja con conjuntos de datos de alta dimensión. La presencia de múltiples variables puede generar redundancia, ruido y dificultades en la visualización e interpretación. En este contexto, algoritmos como la Descomposición en Valores Singulares (SVD), t-Distributed Stochastic Neighbor Embedding (t-SNE), Uniform Manifold Approximation and Projection (UMAP) y el Análisis de Componentes Independientes (ICA) ofrecen diferentes enfoques para representar la información en espacios de menor dimensión, conservando características relevantes del conjunto original.

Cada uno de estos algoritmos se fundamenta en principios matemáticos distintos. Mientras SVD se basa en la factorización matricial y la maximización de varianza, t-SNE y UMAP emplean enfoques no lineales para preservar estructuras locales y topológicas. Por su parte, ICA busca separar señales estadísticamente independientes, lo que lo hace particularmente útil en problemas de procesamiento de señales y análisis biomédico.

El objetivo del presente trabajo es analizar teóricamente y aplicar de manera práctica estos cuatro algoritmos de aprendizaje no supervisado, evaluando su comportamiento, sus diferencias metodológicas y su capacidad para representar estructuras latentes en conjuntos de datos reales. A través de la implementación computacional y la interpretación de resultados, se busca comprender las ventajas, limitaciones y contextos de aplicación de cada técnica dentro del análisis exploratorio de datos.

2.1 SVD

2.1.1 Descripción Teórica

La Descomposición en Valores Singulares (Singular Value Decomposition, SVD) es una técnica fundamental del álgebra lineal aplicada ampliamente en el análisis de datos y aprendizaje automático. Dada una matriz real $A \in \mathbb{R}^{m \times n}$, la SVD permite descomponerla en tres matrices:

$$A = U\Sigma V^T$$

donde Uy Vson matrices ortogonales que contienen los vectores singulares izquierdos y derechos respectivamente, y Σes una matriz diagonal cuyos elementos no negativos corresponden a los valores singulares de A(Strang, 2016).

El objetivo principal de la SVD en el contexto del aprendizaje no supervisado es la reducción de dimensionalidad, permitiendo aproximar una matriz original mediante una representación de menor rango que conserva la mayor cantidad posible de información. Esta propiedad se fundamenta en el teorema de Eckart–Young, el cual establece que la mejor aproximación de rango reducido, en términos de norma de Frobenius, se obtiene truncando la SVD (Eckart & Young, 1936).

Entre sus principales características destacan:

- Puede aplicarse a cualquier matriz real, cuadrada o rectangular.
- No requiere supuestos probabilísticos.
- Permite ordenar la importancia de las componentes según la magnitud de los valores singulares.
- Es numéricamente estable.

En relación con otros métodos de reducción de dimensionalidad, como el Análisis de Componentes Principales (PCA), la SVD constituye una formulación más general. Mientras que PCA se basa en la descomposición de la matriz de covarianza, la SVD se aplica directamente sobre la matriz de datos original. De hecho, PCA puede calcularse utilizando SVD cuando los datos están centrados (James et al., 2021). La diferencia radica principalmente en la interpretación estadística del PCA frente al enfoque algebraico de la SVD.

2.1.2 Usos y aplicaciones

La SVD tiene múltiples aplicaciones en el análisis moderno de datos debido a su capacidad para identificar estructuras latentes.

Principales usos

- Reducción de dimensionalidad
- Compresión de datos
- Eliminación de ruido
- Factorización de matrices incompletas

Áreas de aplicación

1. Sistemas de recomendación

En sistemas como Netflix o Amazon, la SVD se emplea para factorizar matrices usuario–ítem con valores faltantes, permitiendo predecir preferencias mediante modelos de factorización matricial (Koren, Bell, & Volinsky, 2009).

2. Procesamiento de Lenguaje Natural

En el Análisis Semántico Latente (LSA), la SVD se utiliza para reducir matrices término-documento, capturando relaciones semánticas ocultas entre palabras (Deerwester et al., 1990).

3. Visión por computadora

La SVD permite comprimir imágenes reduciendo su rango sin pérdida significativa de calidad visual.

Estas aplicaciones demuestran que la SVD es particularmente adecuada cuando se requiere identificar estructuras subyacentes en datos de alta dimensionalidad.

2.1.3 Dataset utilizado

Para la implementación práctica se utilizó el conjunto de datos **USArrests**, incluido de forma nativa en R. Este dataset contiene estadísticas de criminalidad en los 50 estados de Estados Unidos, con las siguientes variables:

- Murder (tasa de homicidios)
- Assault (tasa de asaltos)
- UrbanPop (porcentaje de población urbana)
- Rape (tasa de violaciones)

2.1.4 Resultados e Interpretación

```
#Aplicación de SVD
```{r}
svd_result <- svd(data_scaled)

# Valores singulares
svd_result$d
```
```

```
[1] 11.024148  6.964086  4.179904  2.915146
```

```
#Varianza explicada
```{r}
var_explained <- svd_result$d^2 / sum(svd_result$d^2)
var_explained

var_acumulada <- cumsum(var_explained)
var_acumulada
```
```

```
[1] 0.62006039 0.24744129 0.08914080 0.04335752
[1] 0.6200604 0.8675017 0.9566425 1.0000000
```

Es posible representar el dataset original de 4 dimensiones utilizando únicamente 2 dimensiones, conservando más del 86% de la información.

```
##Proyección a 2 dimensiones
```{r}
reduced_data <- data.frame(svd_result$u[,1:2])
reduced_data$State <- rownames(USArrests)

head(reduced_data)
```
```

Description: df [6 × 3]

	X1 <dbl>	X2 <dbl>	State <chr>
1	-0.08850212	-0.1611125	Alabama
2	-0.17511901	-0.1525580	Alaska
3	-0.15832905	0.1060383	Arizona
4	0.01269930	-0.1591799	Arkansas
5	-0.22664907	0.2193291	California
6	-0.13600514	0.1403816	Colorado

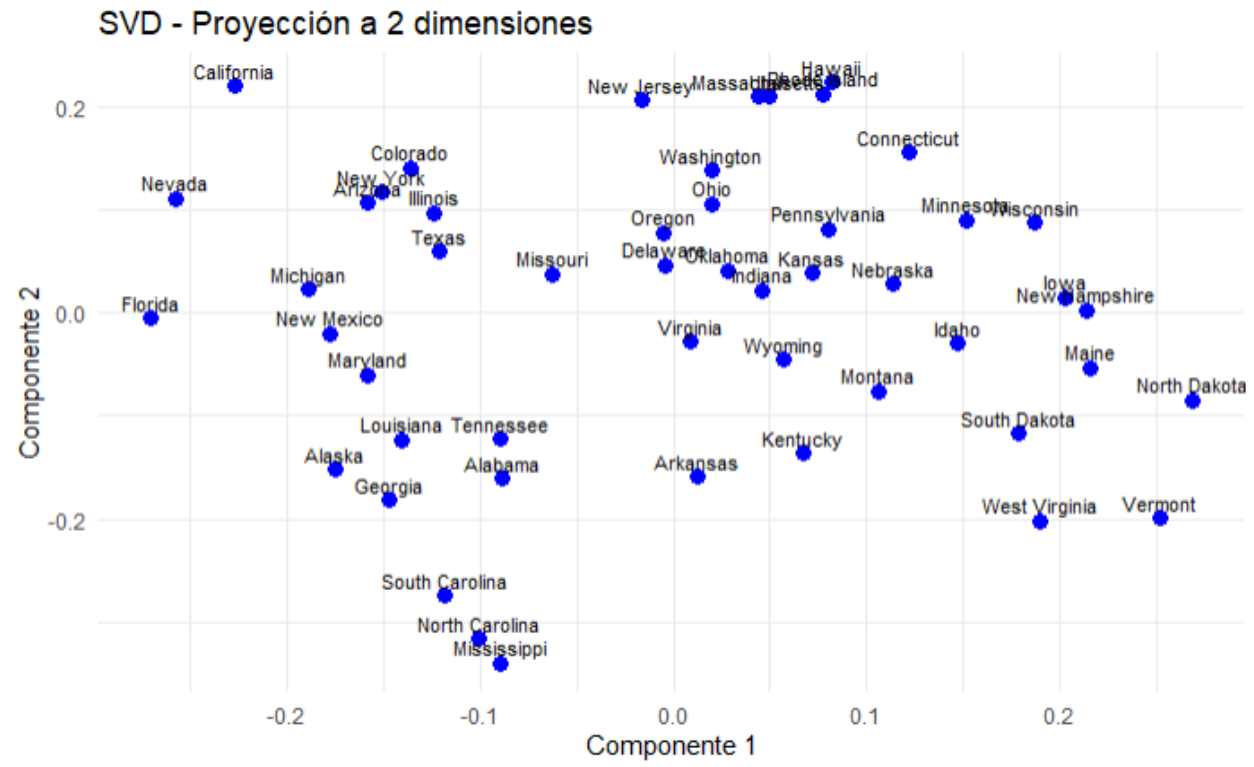
6 rows

```
Componente 1 (X1)

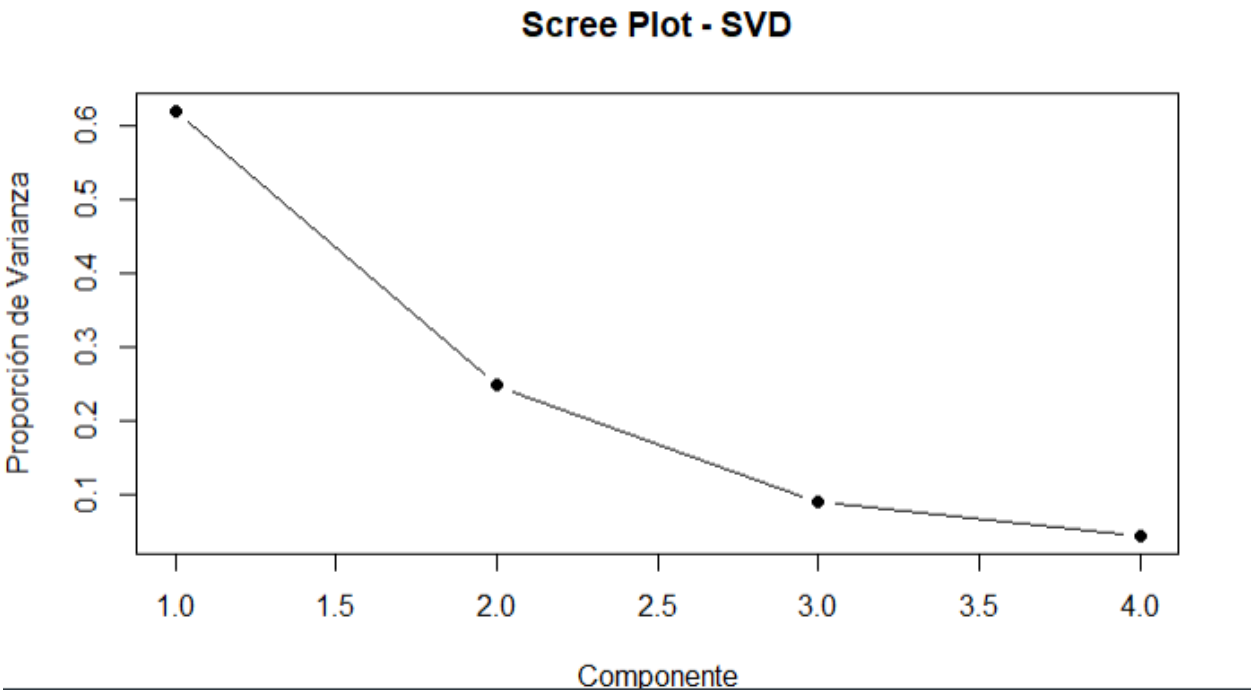
Este eje parece capturar un gradiente general de criminalidad:
  Estados como California y Nevada presentan valores negativos extremos.
  Estados como North Dakota, Vermont y West Virginia aparecen en el extremo opuesto.
Esto sugiere que el primer componente distingue estados con mayores tasas generales de criminalidad frente a estados con menores tasas.

El segundo eje parece capturar variaciones específicas en tipos particulares de delito o en el grado de urbanización.

Por ejemplo:
California presenta un valor alto en X2.
Estados del sur como Mississippi y North Carolina aparecen en valores negativos.
Este eje podría estar diferenciando patrones específicos de criminalidad más que el nivel global.
```




```
En el gráfico se pueden identificar tres tendencias principales:
  Estados con alta criminalidad general.
  Estados con criminalidad moderada.
  Estados con baja criminalidad.
Además, algunos estados del sur tienden a agruparse en zonas similares del espacio reducido, lo que indica
correlaciones estructurales entre sus variables.
```




```
Presencia de un "codo" en el segundo componente, sugiriendo que dos componentes son suficientes para representar
adecuadamente la estructura del conjunto de datos.
```

2.2 t-SNE

2.2.1 Descripción Teórica

El algoritmo t-Distributed Stochastic Neighbor Embedding (t-SNE) es un método no supervisado de reducción de dimensionalidad diseñado principalmente para visualización de datos de alta dimensión (van der Maaten & Hinton, 2008).

A diferencia de métodos lineales como PCA o SVD, t-SNE es un algoritmo **no lineal** que busca preservar la estructura local de los datos. El procedimiento consiste en:

1. Convertir distancias entre puntos en probabilidades condicionales en el espacio original.
2. Definir probabilidades equivalentes en el espacio reducido.
3. Minimizar la divergencia de Kullback-Leibler entre ambas distribuciones.

Una característica distintiva de t-SNE es el uso de una distribución t de Student en el espacio reducido, lo cual reduce el problema conocido como “crowding problem” y mejora la separación visual de clusters (van der Maaten & Hinton, 2008).

El algoritmo depende de un hiperparámetro denominado perplexity, que controla el equilibrio entre preservación local y global.

En comparación con PCA o SVD:

- No maximiza varianza global.
- No es determinista.
- Está optimizado para visualización, no para reconstrucción de datos.

2.2.2 Usos y aplicaciones

t-SNE se utiliza principalmente para análisis exploratorio y visualización.

Principales usos

- Visualización de embeddings
- Exploración de clusters
- Detección de estructuras latentes

Áreas de aplicación

1. Bioinformática

t-SNE es ampliamente utilizado para visualizar datos de expresión génica y clasificar tipos celulares en estudios de secuenciación de ARN (single-cell RNA-seq).

2. Deep Learning

Permite visualizar representaciones internas (embeddings) aprendidas por redes neuronales.

3. Segmentación de clientes

Se emplea para identificar grupos naturales en datos de comportamiento de usuarios.

2.2.3 Dataset utilizado

Se utilizó el dataset **iris**, incluido de forma nativa en R. Contiene 150 observaciones de flores clasificadas en tres especies:

- Setosa
- Versicolor
- Virginica

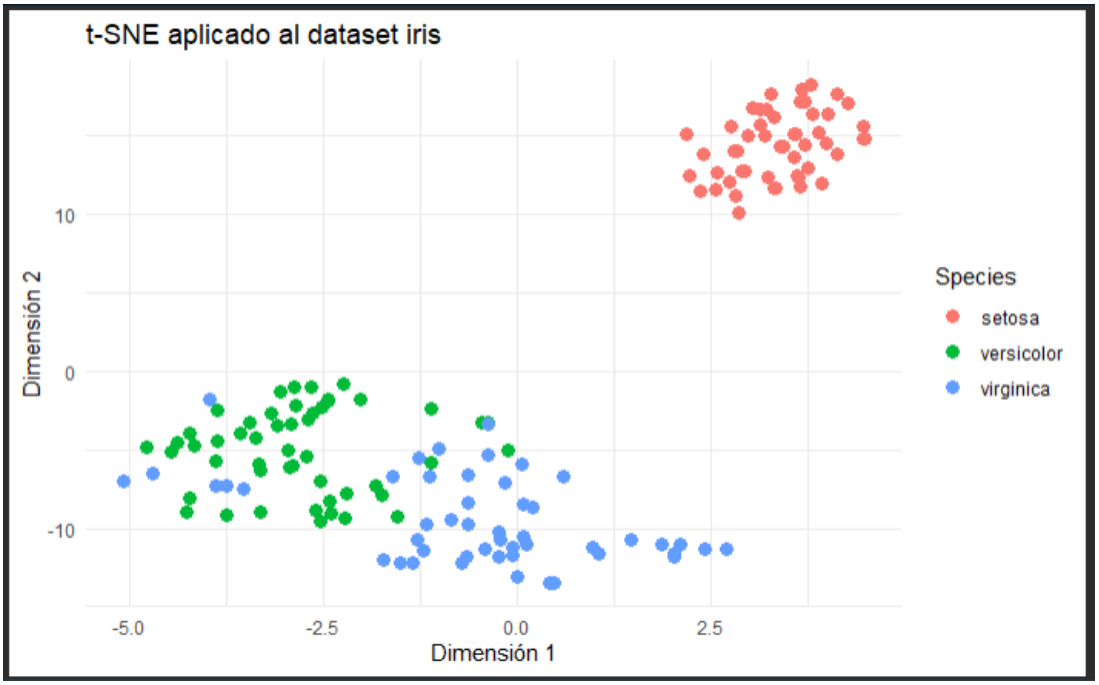
2.2.4 Resultados

```
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
Performing PCA
Read the 150 x 4 data matrix successfully!
OpenMP is working. 1 threads.
Using no_dims = 2, perplexity = 30.000000, and theta = 0.500000
Computing input similarities...
Building tree...
Done in 0.01 seconds (sparsity = 0.711156)!
Learning embedding...
Iteration 50: error is 44.879014 (50 iterations in 0.01 seconds)
Iteration 100: error is 44.742894 (50 iterations in 0.01 seconds)
Iteration 150: error is 45.514841 (50 iterations in 0.01 seconds)
Iteration 200: error is 46.518471 (50 iterations in 0.01 seconds)
Iteration 250: error is 43.671837 (50 iterations in 0.01 seconds)
Iteration 300: error is 0.347849 (50 iterations in 0.01 seconds)
Iteration 350: error is 0.164380 (50 iterations in 0.01 seconds)
Iteration 400: error is 0.159998 (50 iterations in 0.00 seconds)
Iteration 450: error is 0.155128 (50 iterations in 0.00 seconds)
Iteration 500: error is 0.150811 (50 iterations in 0.01 seconds)
Fitting performed in 0.06 seconds.
```

Description: df [6 × 3]

	Dim1 <dbl>	Dim2 <dbl>	Species <fctr>
1	3.579787	15.09327	setosa
2	3.639848	11.76640	setosa
3	2.884961	12.73240	setosa
4	2.737192	12.09248	setosa
5	3.136742	15.63919	setosa
6	3.650022	17.19315	setosa

6 rows



2.2.5 Interpretación

El algoritmo t-SNE convergió adecuadamente tras 500 iteraciones, alcanzando un valor bajo de divergencia de Kullback-Leibler (≈ 0.15), lo que indica que la estructura local del conjunto de datos fue preservada correctamente en la proyección bidimensional.

En la visualización obtenida se observan tres agrupamientos correspondientes a las especies Setosa, Versicolor y Virginica. La especie Setosa aparece claramente separada de las demás, formando un cluster compacto y bien definido. Por su parte, Versicolor y Virginica muestran una ligera superposición, lo que refleja similitudes morfológicas entre ambas especies.

Estos resultados evidencian que t-SNE logra capturar relaciones no lineales y preservar la vecindad local de los datos, generando una representación visual clara de los patrones presentes en el dataset. Sin embargo, las dimensiones obtenidas no poseen interpretación directa y el método no permite reconstruir los datos originales.

2.3 UMAP (Uniform Manifold Approximation and Projection)

2.3.1 Descripción teórica

UMAP es un algoritmo no supervisado de reducción de dimensionalidad basado en fundamentos de geometría diferencial y teoría de grafos (McInnes, Healy, & Melville, 2018). Su objetivo es preservar tanto la estructura local como parte de la estructura global del conjunto de datos.

UMAP parte de tres supuestos principales:

1. Los datos se distribuyen uniformemente sobre una variedad riemanniana.
2. La métrica local puede aproximarse mediante un grafo ponderado.
3. La estructura topológica puede preservarse en un espacio de menor dimensión.

El algoritmo construye un grafo de vecinos más cercanos en el espacio original y luego optimiza una representación de baja dimensión que minimiza una función de pérdida basada en entropía cruzada.

Características principales

- Método no lineal.
- Preserva estructura local y parcialmente global.
- Más rápido que t-SNE en datasets grandes.
- Permite proyecciones estables y reproducibles.

Diferencias con t-SNE

Característica	t-SNE	UMAP
Velocidad	Más lento	Más rápido
Preservación global	Baja	Moderada
Escalabilidad	Limitada	Alta
Base teórica	Probabilística	Topológica

UMAP tiende a conservar mejor la estructura global del dataset y produce representaciones más consistentes.

2.3.2 Usos y aplicaciones

Principales usos

- Visualización de alta dimensión.
- Reducción de dimensionalidad previa a clustering.
- Procesamiento de embeddings.

Áreas de aplicación

1. Bioinformática

Análisis de datos genómicos y transcriptómicos.

2. Procesamiento de imágenes

Reducción de dimensionalidad en features extraídos de redes neuronales.

3. Análisis de comportamiento

Segmentación avanzada de usuarios en plataformas digitales.

2.3.3 Ventajas y limitaciones

Ventajas

- Alta eficiencia computacional.
- Mejor preservación global que t-SNE.
- Escalable a grandes datasets.

Limitaciones

- Sensible a parámetros como n_neighbors.
- Interpretabilidad matemática limitada.

- No permite reconstrucción directa.

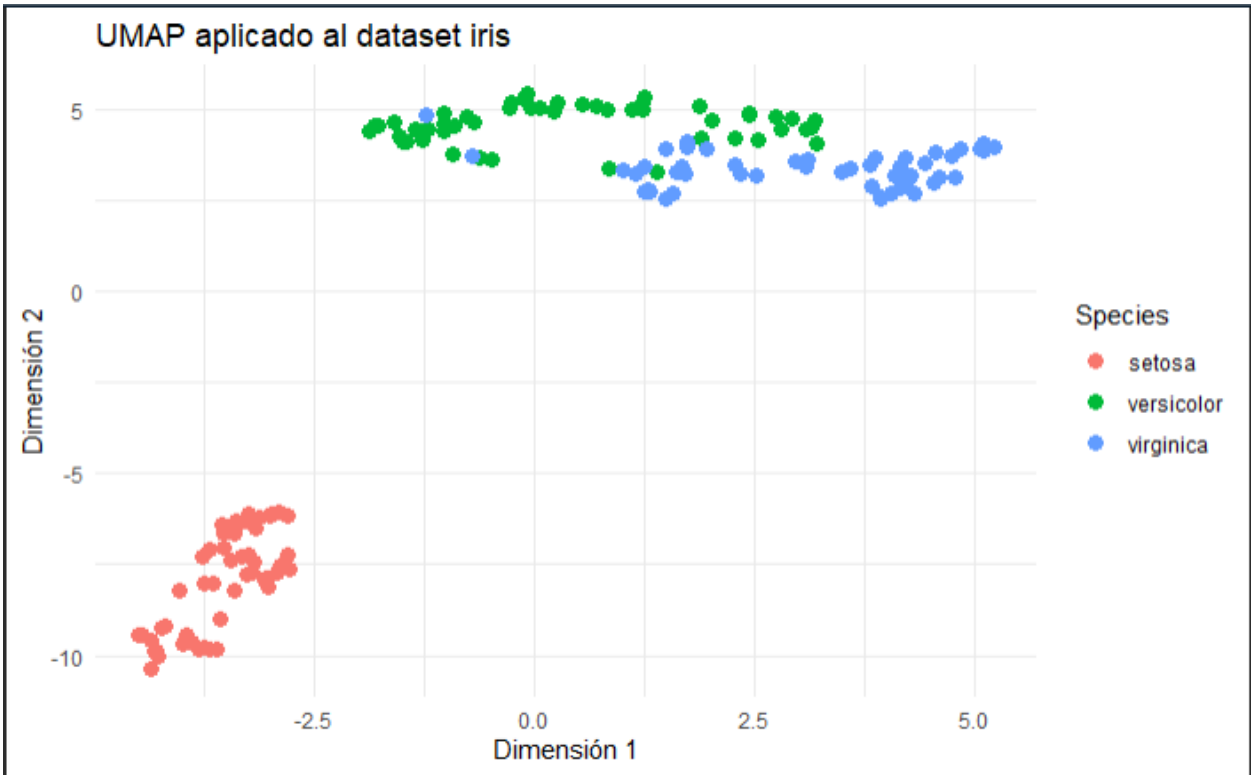
2.3.4 Resultados

```
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Description: df [6 × 3]

	Dim1 <dbl>	Dim2 <dbl>	Species <fctr>
1	-3.184692	-7.412001	setosa
2	-3.751808	-9.758725	setosa
3	-4.232072	-9.224421	setosa
4	-4.339947	-9.589865	setosa
5	-3.525522	-7.034378	setosa
6	-3.113205	-6.190732	setosa

6 rows



2.3.6 Interpretación

La aplicación del algoritmo UMAP al dataset iris permitió obtener una proyección bidimensional que preserva adecuadamente la estructura del conjunto de datos original. En la visualización resultante se observan tres agrupamientos claramente diferenciados correspondientes a las especies Setosa, Versicolor y Virginica.

La especie Setosa aparece completamente separada en la parte inferior del gráfico, formando un cluster compacto y bien definido. Esta separación indica que sus características morfológicas presentan diferencias significativas respecto a las otras especies, lo que confirma la capacidad de UMAP para capturar estructuras locales distintivas.

Por su parte, Versicolor y Virginica se ubican en la parte superior del gráfico y, aunque forman agrupamientos diferenciables, presentan una cercanía mayor entre sí en comparación con Setosa. Esto refleja similitudes en sus características, lo cual es coherente con la estructura conocida del dataset. Sin embargo, a diferencia de t-SNE, UMAP muestra una separación ligeramente más clara entre estas dos especies, lo que sugiere una mejor preservación parcial de la estructura global.

En términos generales, los resultados evidencian que UMAP logra una representación visual clara, con clusters compactos y bien organizados, manteniendo tanto la estructura local como parte de la relación global entre grupos. Esto confirma su utilidad como herramienta eficiente para visualización y análisis exploratorio de datos de dimensionalidad moderada.

2.4 ICA (Independent Component Analysis)

2.4.1 Descripción teórica

El Análisis de Componentes Independientes (ICA) es un método no supervisado cuyo objetivo es separar señales mixtas en componentes estadísticamente independientes (Hyvärinen & Oja, 2000).

A diferencia de PCA o SVD, que buscan componentes no correlacionadas, ICA busca componentes **estadísticamente independientes**, lo cual implica una condición más fuerte.

Matemáticamente, ICA asume que los datos observados Xson combinaciones lineales de fuentes independientes S:

$$X = AS$$

donde:

- Aes la matriz de mezcla.
- Sson las fuentes independientes.

El objetivo es estimar una matriz Wtal que:

$$S = WX$$

Supuestos principales

- Las fuentes son estadísticamente independientes.
- Al menos una de las fuentes es no gaussiana.
- El número de fuentes no excede el número de observaciones.

ICA se fundamenta en la maximización de no-gaussianidad, usualmente mediante curtosis o negentropía.

2.4.2 Diferencias con PCA y SVD

Método	Busca
PCA	Máxima varianza
SVD	Factorización óptima
ICA	Independencia estadística

Mientras PCA elimina correlación lineal, ICA elimina dependencia estadística.

2.4.3 Usos y aplicaciones

Principales usos

- Separación ciega de señales (Blind Source Separation).
- Extracción de señales ocultas.

Áreas de aplicación

1. Procesamiento biomédico

Separación de señales EEG y ECG para eliminar ruido o artefactos (Makeig et al., 1996).

2. Procesamiento de audio

Problema del "cocktail party": separar múltiples voces grabadas simultáneamente.

3. Neurociencia

Identificación de patrones cerebrales independientes.

2.4.4 Ventajas y limitaciones

Ventajas

- Permite recuperar señales originales.
- Útil cuando existe mezcla lineal de fuentes.
- Más potente que PCA para separación de señales.

Limitaciones

- Supone independencia estadística.
- Sensible al ruido.
- Puede presentar ambigüedad de escala y signo.

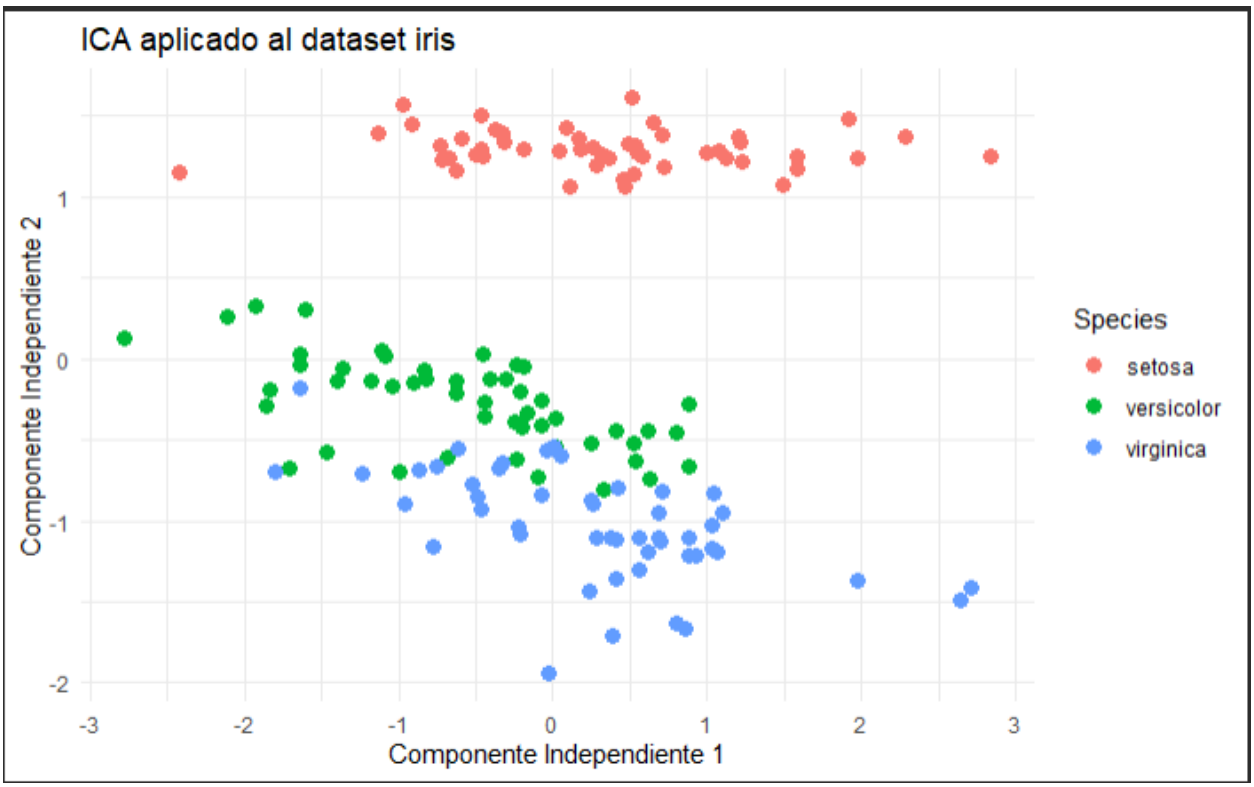
2.4.5 Resultados

```
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Description: df [6 × 3]

	IC1 <dbl>	IC2 <dbl>	Species <fctr>
1	0.5342539	1.313027	setosa
2	-0.6752269	1.234927	setosa
3	-0.3237907	1.392221	setosa
4	-0.5918679	1.360788	setosa
5	0.7104645	1.382004	setosa
6	1.5867856	1.176660	setosa

6 rows



2.4.6 Interpretación

La aplicación del algoritmo ICA al dataset iris permitió obtener dos componentes independientes que representan combinaciones lineales de las variables originales bajo el criterio de máxima independencia estadística. A diferencia de métodos como PCA o SVD, estos componentes no maximizan varianza, sino que buscan separar fuentes subyacentes estadísticamente independientes.

En la visualización resultante se observa una separación clara de la especie Setosa, la cual se agrupa de manera compacta en la parte superior del gráfico, diferenciándose notablemente de las otras dos especies. Esta separación indica que las características morfológicas de Setosa contienen patrones estadísticos independientes que el algoritmo logra aislar eficazmente.

Por otro lado, Versicolor y Virginica aparecen en la región inferior del gráfico, mostrando cierta superposición entre ambas. Aunque se distinguen tendencias diferenciadas —con Virginica mostrando mayor dispersión— la separación no es tan marcada como la observada con UMAP o t-SNE. Esto sugiere que, si bien ICA logra capturar señales independientes relevantes, no está optimizado específicamente para visualización o separación de clusters, sino para descomposición estadística.

En términos generales, los resultados evidencian que ICA permite identificar estructuras latentes independientes dentro de los datos, separando adecuadamente la especie más diferenciada (Setosa) y mostrando relaciones estadísticas entre Versicolor y Virginica. Sin embargo, la separación visual

es menos pronunciada en comparación con métodos no lineales, lo cual es consistente con la naturaleza lineal del modelo ICA.

Link github: https://github.com/dannyrmrz/Tarea2-MD.git

Referencias

Hyvärinen, A., & Oja, E. (2000). Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4–5), 411–430.

Kobak, D., & Berens, P. (2019). The art of using t-SNE for single-cell transcriptomics. *Nature Communications*, 10(1), 5416.

McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*.

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9, 2579–2605.