

UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

Tecnologías de la información



Extracción de conocimiento en base de datos

IV.1. Algoritmos de agrupación (25%)

IDGS91N

Presenta:

Brahn Raudales

Docente:

Enrique Mascote

sábado, 29 de noviembre de 2025

1. Introducción

El análisis no supervisado es un componente fundamental dentro de los procesos de *extracción de conocimiento*, ya que permite identificar patrones, relaciones, estructuras y distribuciones ocultas dentro de los datos sin necesidad de etiquetas o valores conocidos de salida. Dos de las técnicas más importantes en este ámbito son el **clustering** y la **reducción de dimensionalidad**.

El **clustering** permite agrupar objetos similares según su comportamiento o características, siendo útil para segmentar clientes, detectar patrones ocultos o encontrar grupos en grandes bases de datos. Por otro lado, la **reducción de dimensionalidad** transforma conjuntos de datos de alta dimensión en representaciones más compactas, manteniendo la mayor cantidad posible de información relevante. Esto facilita la visualización, mejora el rendimiento de los modelos y reduce el ruido.

El objetivo de este reporte es **describir los principales algoritmos de agrupación y reducción de dimensionalidad, ejemplificar su funcionamiento y comparar sus aplicaciones prácticas**, reforzando su utilidad dentro del análisis exploratorio de datos.

2. Algoritmos de agrupación (Clustering)

A continuación, se describen tres algoritmos ampliamente utilizados: **K-Means**, **Clustering Jerárquico** y **DBSCAN**.

2.1 K-Means

Principio de funcionamiento

K-Means busca dividir los datos en K grupos (clusters) mediante la minimización de la distancia entre los puntos de un cluster y su centroide.

El algoritmo sigue estos pasos:

1. Seleccionar K centroides iniciales.
2. Asignar cada punto al centroide más cercano (distancia euclidiana).
3. Recalcular los centroides.
4. Repetir hasta convergencia.

Parámetros clave

- **K:** número de clusters deseados.
- **Número de iteraciones máximas.**
- **Método de inicialización (k-means++).**

Ventajas

- Simple y rápido.
- Escalabilidad para grandes volúmenes de datos.
- Intuitivo y fácil de interpretar.

Limitaciones

- Requiere definir K previamente.
- Sensible a outliers.
- No funciona bien con clusters de forma irregular o de densidades distintas.

Ejemplo simple (pseudocódigo)

Inicializar K centroides aleatorios

Mientras no converja:

 Para cada punto:

 Asignar al centroide más cercano

 Recalcular centroides

Retornar clusters

2.2 Clustering Jerárquico (Aglomerativo)

Principio de funcionamiento

El clustering aglomerativo construye una jerarquía de grupos mediante fusiones sucesivas:

1. Cada punto inicia como un cluster independiente.
2. Se unen los dos clusters más cercanos según un criterio de distancia.
3. Continúa hasta formar un solo cluster o alcanzar el número deseado.

Se visualiza mediante un **dendrograma**.

Parámetros clave

- **Métrica de distancia:** euclidiana, Manhattan, cosine.
- **Método de enlace:** single, complete, average, ward.
- **Número de clusters deseados (opcional).**

Ventajas

- No requiere especificar K al inicio.
- El dendrograma permite análisis visual profundo.
- Capaz de encontrar estructuras jerárquicas.

Limitaciones

- Alto costo computacional para grandes datasets.
- Sensible al ruido.
- Las fusiones no se pueden deshacer (greedy).

Pseudocódigo

Inicializar cada punto como un cluster individual

Mientras número de clusters > 1:

 Calcular distancia entre clusters

 Unir los dos más cercanos

Generar dendrograma

2.3 DBSCAN

Principio de funcionamiento

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) agrupa datos según densidad:

- Puntos en regiones densas forman clusters.
- Puntos en regiones aisladas se consideran *ruido* (outliers).

Clasifica puntos en:

- **Core points** (densidad alta)
- **Reachable points**
- **Noise**

Parámetros clave

- **eps:** radio máximo de vecindad.
- **min_samples:** puntos mínimos para considerar una región como "densa".

Ventajas

- Detecta clusters de forma arbitraria.
- Identifica outliers automáticamente.
- No requiere decidir K.

Limitaciones

- Sensible a la elección de eps y min_samples.
- Menos efectivo con densidades muy variables.
- No escala tan bien como K-Means.

Ejemplo simple (diagrama textual)

Para cada punto:

Si vecindad \geq min_samples \rightarrow core point

Expandir cluster a puntos alcanzables

Puntos no alcanzables \rightarrow ruido

3. Algoritmos de reducción de dimensionalidad

Se describen **PCA** y **t-SNE**, dos métodos ampliamente utilizados en análisis exploratorio.

3.1 PCA (Análisis de Componentes Principales)

Fundamento matemático

PCA proyecta los datos en un nuevo espacio formado por **componentes principales**, que maximizan la varianza.

Matemáticamente:

1. Se calcula la matriz de covarianza.
2. Se obtienen los valores y vectores propios.
3. Los vectores propios principales forman los nuevos ejes.

$$PC = X \cdot W$$

Parámetros clave

- **n_components:** número de dimensiones finales.
- **Whitening (opcional).**

Ventajas

- Reduce ruido.
- Mejora tiempos de cómputo.
- Facilita visualización en 2D o 3D.

Limitaciones

- Solo captura relaciones lineales.
- Puede perder interpretabilidad.

Ejemplo simple

Estandarizar datos

Calcular matriz de covarianza

Calcular vectores propios

Elegir los de mayor varianza

Transformar datos

3.2 t-SNE (t-distributed Stochastic Neighbor Embedding)

Fundamento conceptual

t-SNE convierte similitudes entre puntos en probabilidades y luego crea un mapa de baja dimensión preservando vecindades locales.

- Coloca puntos similares cerca.
- Separa claramente grupos distintos.

Parámetros clave

- **perplexity:** controla densidad local.
- **learning_rate:** velocidad de ajuste.
- **n_iter:** iteraciones.

Ventajas

- Excelente para visualizar datos de alta dimensión.
- Descubre estructuras complejas.
- Preserva relaciones locales.

Limitaciones

- No escala bien.
- No sirve para modelos predictivos.
- Resultado cambia entre ejecuciones.

Ejemplo ilustrativo

Calcular similitudes en alta dimensión

Inicializar puntos aleatorios en 2D

Minimizar divergencia KL entre espacios

Mostrar mapa final 2D

4. Comparativa y conclusiones

Comparativa general

Técnica	Objetivo	Cuándo usar
Clustering (K-Means, Jerárquico, DBSCAN)	Agrupar datos similares	Cuando se desea segmentar, descubrir patrones o explorar estructura interna
Reducción de dimensionalidad (PCA, t-SNE)	Simplificar y visualizar datos	Cuando los datos tienen muchas dimensiones y se quiere reducir ruido o visualizar

Conclusiones

- El clustering permite **identificar grupos naturales en los datos** sin etiquetas, siendo útil en segmentación, análisis exploratorio y detección de patrones.
- La reducción de dimensionalidad permite **visualizar datos complejos, reducir ruido y mejorar modelos posteriores**.
- K-Means es eficiente, DBSCAN detecta formas arbitrarias y Clustering Jerárquico genera estructuras interpretables.
- PCA es ideal para simplificar datos lineales, mientras que t-SNE ofrece visualizaciones ricas para datos no lineales.
- Ambas técnicas son complementarias: primero puede usarse reducción de dimensionalidad, y después clustering, o viceversa, dependiendo del problema.

5. Referencias

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.). O'Reilly Media.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning: With Applications in R* (2nd ed.). Springer.

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

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

OpenAI. (2025). ChatGPT [Modelo de lenguaje]. OpenAI. <https://chat.openai.com/>