UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

Desarrollo de software

Extracción de Conocimiento en Bases de Datos

IV.1. Algoritmos de agrupación (25%)

IDGS91N

PRESENTA:

Juan Carlos Medina Sánchez

DOCENTE:

Enrique Mascote

Chihuahua, Chih., 29 de noviembre de 2025

1. Introducción

En el campo de la ciencia de datos y la minería de información, existen tareas que requieren analizar grandes cantidades de datos sin la presencia de etiquetas o categorías predeterminadas. Entre las técnicas más utilizadas para este propósito se encuentran los algoritmos de agrupación (clustering) y los métodos de reducción de dimensionalidad.

El clustering permite identificar patrones o estructuras internas en los datos, formando grupos basados en similitud. Es ampliamente utilizado cuando se busca detectar segmentos de clientes, identificar comportamientos anómalos o explorar datos sin conocer previamente sus clases.

La reducción de dimensionalidad, por otra parte, transforma conjuntos de datos con muchas variables en representaciones más compactas, preservando la información más relevante. Estas técnicas son útiles cuando los datos tienen alta dimensionalidad, lo que puede generar problemas de sobreajuste, tiempo de cómputo elevado o dificultad para visualizar patrones.

En conjunto, ambos enfoques permiten extraer conocimiento relevante, explorar estructuras internas y preparar datos para modelos más complejos. Este reporte presenta tres algoritmos de clustering y dos algoritmos de reducción de dimensionalidad, describiendo su funcionamiento, parámetros clave, ventajas, limitaciones y ejemplos de aplicación.

2. Algoritmos de Agrupación (Clustering)

A continuación, se describen tres algoritmos: K-means, Clustering jerárquico aglomerativo, y DBSCAN.

2.1. K-means

Principio de funcionamiento

K-means es un algoritmo de clustering basado en distancias. Su objetivo es dividir los datos en k grupos de manera que cada punto pertenezca al clúster cuyo centroide esté más cercano.

Proceso básico:

1. Se eligen aleatoriamente k centroides iniciales.
2. Cada punto se asigna al centroide más cercano (formación de clústeres).
3. Se recalculan los centroides como el promedio de los puntos asignados.

4. Repetir pasos 2–3 hasta que los centroides ya no cambien significativamente.

Parámetros clave

- **k**: número de clústeres.
- **Método de inicialización** (k-means++, random).
- **Criterio de convergencia** (cambios mínimos entre iteraciones).
- **Distancia** (euclidiana por defecto).

Ventajas

- Simple, eficiente y fácil de implementar.
- Escala bien con grandes volúmenes de datos.
- Produce clústeres compactos y claramente definidos.

Limitaciones

- Necesita definir k antes de entrenar.
- Sensible a outliers.
- No funciona bien con clústeres no esféricos o con densidades diferentes.

Ejemplo (pseudocódigo)

```
Elegir k centroides iniciales
REPETIR:
    Para cada punto:
        asignar al centroide más cercano
    Recalcular centroides
HASTA que no cambien los centroides
```

2.2. Clustering jerárquico aglomerativo

Principio de funcionamiento

El clustering jerárquico construye una estructura en forma de árbol (dendrograma), combinando progresivamente puntos o grupos.

Proceso aglomerativo:

1. Cada punto inicia como un clúster independiente.
2. Se calcula la distancia entre todos los clústeres.
3. Se unen los dos clústeres más cercanos.
4. Repetir hasta obtener un único clúster que lo contiene todo.

Parámetros clave

- **Método de enlace**:
    - Single-linkage: distancia mínima
    - Complete-linkage: distancia máxima
    - Average-linkage: promedio
    - Ward: minimización de varianza
- **Métrica de distancia**: euclidiana, Manhattan, Minkowski.
- **Número de clústeres a cortar en el dendrograma**.

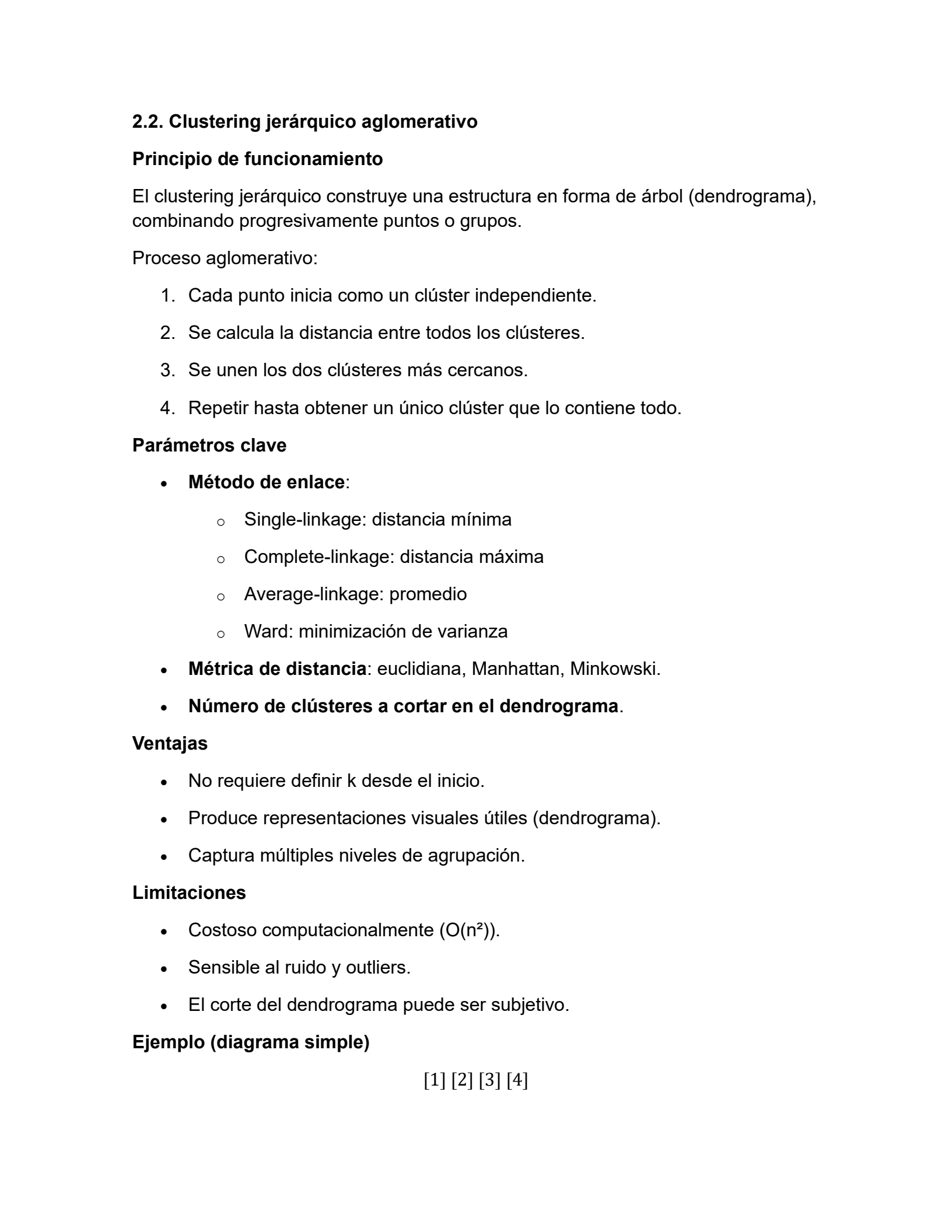
Ventajas

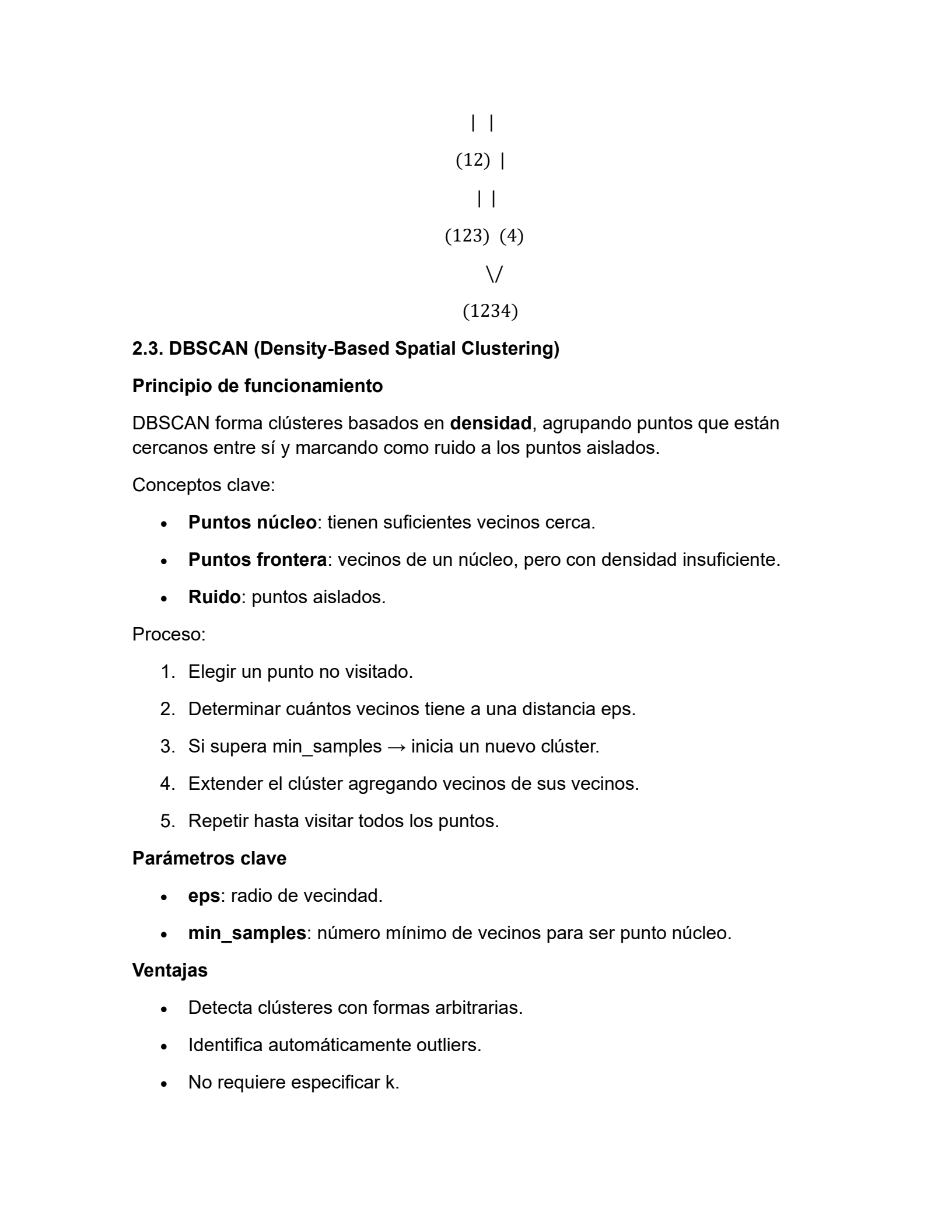
- No requiere definir k desde el inicio.
- Produce representaciones visuales útiles (dendrograma).
- Captura múltiples niveles de agrupación.

Limitaciones

- Costoso computacionalmente ($O(n^2)$).
- Sensible al ruido y outliers.
- El corte del dendrograma puede ser subjetivo.

Ejemplo (diagrama simple)





2.3. DBSCAN (Density-Based Spatial Clustering)

Principio de funcionamiento

DBSCAN forma clústeres basados en **densidad**, agrupando puntos que están cercanos entre sí y marcando como ruido a los puntos aislados.

Conceptos clave:

- **Puntos núcleo**: tienen suficientes vecinos cerca.
- **Puntos frontera**: vecinos de un núcleo, pero con densidad insuficiente.
- **Ruido**: puntos aislados.

Proceso:

1. Elegir un punto no visitado.
2. Determinar cuántos vecinos tiene a una distancia eps.
3. Si supera min_samples → inicia un nuevo clúster.
4. Extender el clúster agregando vecinos de sus vecinos.
5. Repetir hasta visitar todos los puntos.

Parámetros clave

- **eps**: radio de vecindad.
- **min_samples**: número mínimo de vecinos para ser punto núcleo.

Ventajas

- Detecta clústeres con formas arbitrarias.
- Identifica automáticamente outliers.
- No requiere especificar k.

Limitaciones

- Muy sensible a eps.
- Difícil de usar en datos de alta dimensión.

Ejemplo (pseudocódigo)

Para cada punto:

Si no visitado:

Obtener vecinos

Si vecinos >= min_samples:

crear nuevo cluster

expandir buscándolo recursivamente

Si no:

marcar como ruido

3. Algoritmos de Reducción de Dimensionalidad

3.1. PCA (Análisis de Componentes Principales)

Fundamento matemático

PCA transforma los datos a un nuevo sistema de ejes que maximiza la varianza. Usa álgebra lineal:

1. Estandarizar los datos.
2. Calcular la **matriz de covarianzas**.
3. Obtener **autovalores y autovectores**.
4. Ordenar los autovectores por varianza explicada.
5. Proyectar los datos en los componentes principales.

Parámetros clave

- Número de componentes (n_components).
- Método de normalización.
- Varianza mínima requerida.

Ventajas

- Reduce dimensionalidad manteniendo información esencial.
- Acelera entrenamiento de modelos.
- Facilita visualización de datos.

Limitaciones

- Supone relaciones lineales.
- Componentes pueden ser difíciles de interpretar.
- Sensible a escala.

Ejemplo ilustrativo

Datos originales: (x1, x2, x3)

↓ Transformación lineal

Nuevos ejes: PC1, PC2

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

Fundamento conceptual

t-SNE reduce dimensionalidad preservando distancias **locales**.
Es ideal para visualizaciones 2D o 3D.

Proceso:

1. Convierte distancias en probabilidades a nivel alto (original).
2. Convierte distancias en probabilidades en baja dimensión.
3. Minimiza la divergencia entre ambas distribuciones.

Parámetros clave

- perplexity (entre 5–50)
- learning_rate
- número de iteraciones
- n_components (normalmente 2)

Ventajas

- Excelente para visualizar clústeres.

- Detecta estructuras locales y subgrupos.

Limitaciones

- Alto costo computacional.
- No preserva estructura global.
- No sirve como método para modelado predictivo.

Ejemplo ilustrativo

Datos de 100 dimensiones

↓ t-SNE

Mapa 2D donde aparecen clústeres visibles

4. Comparativa y Conclusiones

Clustering vs. Reducción de dimensionalidad

Aspecto	**Clustering**	**Reducción de dimensionalidad**
Objetivo	Crear grupos	Resumir variables
Tipo de aprendizaje	No supervisado	No supervisado
Salida	Etiquetas o grupos	Nuevos ejes o representaciones
Uso común	Segmentación	Visualización, preprocesamiento
Detecta outliers	Solo DBSCAN	No directamente

Situaciones prácticas

- **Usar clustering cuando:**
  - Quiero segmentar clientes.
  - Deseo detectar grupos internos.
  - Busco encontrar anomalías (DBSCAN).
- **Usar reducción de dimensionalidad cuando:**
  - Los datos tienen demasiadas variables.
  - Quiero visualizar datos en 2D.
  - Necesito mejorar eficiencia de un modelo.

Conclusiones

Los algoritmos de agrupación permiten explorar y segmentar datos sin etiquetas, revelando estructuras ocultas. K-means es eficiente para clústeres esféricos, el clustering jerárquico es útil para entender la estructura de los datos, y DBSCAN sobresale en detectar outliers y clústeres de formas irregulares.

Los métodos de reducción de dimensionalidad como PCA y t-SNE permiten trabajar con datos complejos, eliminando redundancia y facilitando visualización. PCA preserva variabilidad global, mientras que t-SNE revela relaciones locales.

El uso combinado de clustering y reducción de dimensionalidad es común en análisis exploratorio avanzado y resulta fundamental en la extracción de conocimiento en grandes volúmenes de datos.

Referencias (APA)

- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn*. O'Reilly Media.
- Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9, 2579–2605.