UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

Tecnologías de la información

Extracción de Conocimiento en Bases de Datos

IV.1. Algoritmos de agrupación

Docente

Enrique Mascote

Alumno

Myriam Raquel Almuina Orozco

IDGS 91N

Sábado, 29 de noviembre del 2025

1. Introducción

En la extracción de conocimiento, los algoritmos de aprendizaje no supervisado permiten descubrir patrones ocultos en datos sin etiquetas. Entre sus funciones principales destacan:

- **Agrupación (clustering):** identificar grupos naturales dentro de un conjunto de datos, como clientes con comportamientos similares.

- **Reducción de dimensionalidad:** simplificar los datos manteniendo la mayor parte de su información, lo que mejora la visualización, reduce ruido y acelera cálculos.

Estos métodos son fundamentales en análisis exploratorio, segmentación, visualización y como paso previo a otros algoritmos de aprendizaje.

2. Algoritmos de agrupación (Clustering)

2.1 K-Means

Principio de funcionamiento:
K-Means divide los datos en *k* grupos. Para ello:

1. Selecciona *k* centroides iniciales.

2. Asigna cada punto al centroide más cercano.

3. Recalcula los centroides como el promedio de cada grupo.

4. Repite hasta que los centroides no cambien.

Parámetros clave:

- k: número de clusters.

- max_iter: iteraciones máximas.

- init: método de inicialización (k-means++ recomendado).

Ventajas:

- Rápido y eficiente en grandes datasets.
- Fácil de implementar y de interpretar.

Limitaciones:

- Requiere definir *k*.
- Sensible a outliers.
- Solo modela clusters esféricos.

Ejemplo simple (pseudocódigo):

```
Inicializar k centroides
Repetir:
   Para cada punto:
      Asignarlo al centroide más cercano
   Recalcular centroides
Hasta convergencia
```

2.2 Clustering jerárquico (Aglomerativo)

Principio de funcionamiento:
Construye una jerarquía de agrupamientos:

1. Cada punto inicia como un cluster individual.
2. En cada paso se unen los dos clusters más cercanos.
3. Se detiene cuando queda un solo cluster o cuando se elige un número deseado de grupos.

Parámetros clave:

- linkage: método de unión (single, complete, average, ward).
- distance_metric: métrica de distancia (Euclidean, Manhattan).
- n_clusters: número final de grupos.

Ventajas:

- No requiere definir *k* inicialmente.
- Permite visualizar la estructura mediante dendrogramas.

Limitaciones:

- Costoso en datasets grandes ($O(n^2)$).
- Sensible al ruido dependiendo del linkage.

Ejemplo visual:
Un dendrograma donde las ramas muestran cómo se unen los clusters.

2.3 DBSCAN

Principio de funcionamiento:
Encuentra clusters basados en densidad:

- Si un punto tiene suficientes vecinos cercanos, es un punto central.
- Los puntos conectados a centros forman un cluster.
- Puntos aislados son clasificados como *ruido*.

Parámetros clave:

- eps: radio de vecindad.
- min_samples: número mínimo de puntos para ser considerado un cluster.

Ventajas:

- Detecta formas arbitrarias de clusters.
- Identifica ruido/outliers de forma natural.
- No requiere definir *k*.

Limitaciones:

- Difícil elegir eps.
- En datos con densidad variable puede fallar.

Pseudocódigo simple:

```
Para cada punto no visitado:
   Marcar como visitado
   Encontrar vecinos dentro de eps
   Si hay suficientes (>= min_samples):
      Expandir cluster
   Si no:
      Marcar como ruido
```

3. Algoritmos de reducción de dimensionalidad

3.1 PCA (Principal Component Analysis)

Fundamento matemático/conceptual:
PCA transforma los datos a nuevas dimensiones que:

- Capturan la mayor varianza.
- Son combinaciones lineales de las características originales.
- Están ordenadas por importancia (componente 1, componente 2…).

Pasos matemáticos básicos:

1. Estandarizar los datos.
2. Calcular matriz de covarianza.
3. Obtener eigenvalues/eigenvectors.
4. Ordenar y proyectar los datos en los componentes principales.

Parámetros clave:

- n_components: cuántos componentes conservar.
- svd_solver: algoritmo usado para descomposición.

Ventajas:

- Reduce dimensiones conservando variabilidad.
- Mejora rendimiento en modelos.
- Ayuda a visualizar en 2D/3D.

Limitaciones:

- Solo captura relaciones lineales.
- Componentes no siempre son interpretables.

Ejemplo simple:
Reducir un dataset de 20 variables a 2 componentes para visualizar clusters.

3.2 t-SNE

Fundamento conceptual:
t-SNE reduce dimensiones preservando relaciones de proximidad:

- Los puntos cercanos permanecen juntos.
- Los lejanos se separan claramente.
- Ideal para visualizar clusters complejos en 2D.

Parámetros clave:

- perplexity: controla balance entre vecinos locales/globales (20–50 recomendado).
- learning_rate.

- n_iter.

Ventajas:

- Excelentes visualizaciones.
- Capta estructuras no lineales.

Limitaciones:

- Lento en datasets grandes.
- No conserva distancias globales.
- No sirve para modelado, solo visualización.

Ejemplo:
Visualizar datos de clientes y ver grupos naturales que K-Means también puede descubrir.

4. Comparativa y conclusiones

Técnica	Objetivo	Cuándo usarla	Ventajas principales
Clustering (K-Means, Jerárquico, DBSCAN)	Agrupar datos sin etiquetas	Segmentación de clientes, patrones ocultos, análisis exploratorio	Descubre grupos naturales
Reducción de dimensionalidad (PCA, t-SNE)	Simplificar datos con mínima pérdida de información	Visualización, preprocesamiento, eliminación de ruido	Reduce complejidad y mejora modelos

Conclusión general:
Los algoritmos de clustering permiten identificar grupos y patrones ocultos, mientras que la reducción de dimensionalidad ayuda a visualizar y preparar datos de manera más eficiente. Ambos se complementan: técnicas como PCA pueden usarse antes de clustering para mejorar resultados.

5. Referencias (APA)

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques*. Morgan Kaufmann.

Pedregosa, F. et al. (2011). *Scikit-Learn: Machine learning in Python*. Journal of Machine Learning Research.

Van der Maaten, L., & Hinton, G. (2008). *Visualizing data using t-SNE*. Journal of Machine Learning Research.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.