Universidad Tecnológica de chihuahua

Tecnologías de la información

Extracción de Conocimiento en Bases de Datos

Enrique Mascote

IV.1. Algoritmos de agrupación

Marco Duarte – IDGS91N

1. Introducción

En el área de minería de datos y aprendizaje automático, el análisis no supervisado permite descubrir patrones ocultos sin necesidad de etiquetas predefinidas. Dos de las técnicas más importantes dentro de este enfoque son el agrupamiento (clustering) y la reducción de dimensionalidad. El clustering permite identificar grupos naturales dentro de los datos, facilitando la segmentación de clientes, la detección de anomalías o la comprensión de estructuras subyacentes. Por otro lado, la reducción de dimensionalidad busca simplificar conjuntos de datos complejos preservando la mayor cantidad de información posible, lo cual es crucial cuando se trabaja con datos de alta dimensionalidad que pueden dificultar la visualización o el modelado.

Ambas técnicas son fundamentales en la extracción de conocimiento porque permiten organizar, sintetizar y visualizar patrones en los datos, especialmente cuando no existe una variable objetivo. En este reporte se describen tres algoritmos de clustering y dos métodos de reducción de dimensionalidad, profundizando en su funcionamiento, parámetros clave, ventajas y limitaciones, así como ejemplos ilustrativos de su aplicación.

2. Algoritmos de Agrupación (Clustering)

A continuación, se describen tres técnicas de agrupación: K-means, Clustering Jerárquico Aglomerativo y DBSCAN.

2.1 K-means

Principio de funcionamiento

K-means busca dividir los datos en k grupos mediante la minimización de la distancia entre cada punto y el centroide del cluster. El algoritmo sigue un proceso iterativo:

Elegir k centroides iniciales.

Asignar cada punto al centroide más cercano (distancia Euclidiana).

Recalcular los centroides como el promedio de los puntos asignados.

Repetir los pasos 2–3 hasta que los centroides estabilicen.

Parámetros clave

k: número de clusters.

Método de inicialización: por ejemplo, k-means++.

Número máximo de iteraciones.

Ventajas

Simple y rápido.

Escalable a grandes volúmenes de datos.

Fácil de implementar e interpretar.

Limitaciones

Requiere definir k antes de iniciar.

Sensible a valores atípicos.

No funciona bien con clusters no esféricos o con diferente densidad.

Ejemplo (pseudocódigo)

Inicializar k centroides al azar

Repetir:

  Para cada punto:

    Asignar al centroide más cercano

  Recalcular centroides como media de puntos asignados

Hasta convergencia

2.2 Clustering Jerárquico Aglomerativo

Principio de funcionamiento

Forma una estructura jerárquica de clusters mediante un enfoque ascendente (aglomerativo):

Cada punto inicia como un cluster individual.

Se fusionan los dos clusters más cercanos según un criterio (linkage).

Este proceso continúa hasta obtener un único cluster.

El resultado puede visualizarse mediante un dendrograma.

Parámetros clave

Método de enlace: single, complete, average, ward.

Métrica de distancia.

Altura de corte para determinar el número final de clusters.

Ventajas

Muy interpretativo gracias al dendrograma.

No requiere especificar k al inicio.

Funciona bien con diferentes formas de clusters.

Limitaciones

Costoso computacionalmente para grandes datasets.

Una vez fusionados, los clusters no se pueden deshacer.

Ejemplo simple

Un diagrama de flujo básico:

Inicio → Calcular matriz de distancias → Unir clusters más cercanos →

Actualizar matriz → Repetir hasta 1 cluster → Generar dendrograma

2.3 DBSCAN

Principio de funcionamiento

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) agrupa puntos según regiones de alta densidad y marca como ruido los puntos aislados.

Se basa en dos conceptos:

Epsilon (ε): distancia máxima para considerar vecinos.

MinPts: mínimo de puntos para formar un cluster denso.

Clasifica los puntos en:

Core points (densidad suficiente)

Border points

Noise points (ruido)

Parámetros clave

ε (epsilon)

MinPts

Ventajas

Detecta clusters de forma arbitraria.

Maneja ruido y outliers.

No requiere especificar número de clusters.

Limitaciones

Sensibilidad a la elección de ε.

Difícil de ajustar en alta dimensionalidad.

Ejemplo

Para cada punto no visitado:

Marcar como visitado

Obtener vecinos dentro de ε

Si vecinos >= MinPts:

Crear nuevo cluster expandiendo vecinos

Si no, marcar como ruido

3. Algoritmos de Reducción de Dimensionalidad

Se analizan dos técnicas: PCA y t-SNE.

3.1 Análisis de Componentes Principales (PCA)

Fundamento matemático

PCA transforma un conjunto de variables correlacionadas en nuevas variables no correlacionadas llamadas componentes principales, que son combinaciones lineales de las originales.

Matemáticamente:

Estandarizar los datos.

Calcular la matriz de covarianza.

Obtener vectores y valores propios.

Ordenar componentes por varianza explicada.

Parámetros clave

Número de componentes (n_components).

Método de estandarización previo.

Ventajas

Reduce dimensionalidad preservando varianza.

Acelera entrenamientos.

Facilita visualización 2D o 3D.

Limitaciones

Es lineal; no captura relaciones no lineales.

Las componentes no siempre son interpretables.

Ejemplo simple

Estandarizar datos →

Calcular matriz de covarianza →

Obtener eigenvectors →

Elegir componentes principales →

Transformar datos

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

Fundamento conceptual

t-SNE es una técnica no lineal que proyecta datos de alta dimensionalidad a 2D o 3D manteniendo relaciones de vecindad. Optimiza una función de divergencia entre distribuciones de proximidad en el espacio original y en el reducido.

Parámetros clave

Perplexity: controla equilibrio entre estructuras locales y globales.

Learning rate.

Número de iteraciones.

Ventajas

Excelente para visualizar agrupaciones ocultas.

Capta relaciones no lineales.

Limitaciones

Alto costo computacional.

No preserva estructura global.

No sirve para modelos posteriores (solo visualización).

Ejemplo conceptual

Calcular similitudes entre puntos →

Inicializar posiciones 2D →

Optimizar divergencia →

Obtener representación reducida

4. Comparativa y Conclusiones

Cuándo usar clustering vs reducción de dimensionalidad

Objetivo	Clustering	Reducción de Dimensionalidad
Encontrar grupos	✕	
Visualizar datos	(parcial)	
Acelerar modelos	✕	
Detectar anomalías		✕
Preprocesamiento	✕	

Situaciones prácticas

Clustering se usa cuando se quiere segmentar, detectar patrones o descubrir grupos naturales (clientes, documentos, imágenes).

Reducción de dimensionalidad se utiliza antes del clustering, visualización o entrenamiento para simplificar datos complejos.

Conclusiones

El clustering y la reducción de dimensionalidad son técnicas esenciales para extraer conocimiento en entornos no supervisados. Mientras que el clustering identifica patrones grupales, la reducción de dimensionalidad permite simplificar y visualizar los datos conservando información relevante. Su uso combinado suele ofrecer los mejores resultados, especialmente en datasets complejos y de alta dimensionalidad.

5. Referencias (APA)

Geron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow. O'Reilly.

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

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, 9, 2579–2605.