UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

TECNOLOGÍAS DE LA INFORMACIÓN

EXTRACCIÓN DE CONOCIMIENTO EN BASES DE DATOS

IV.1. ALGORITMOS DE AGRUPACIÓN

IDGS91N

PRESENTA:

REGINA CHÁVEZ TAMAYO - 6521110019

DOCENTE:

LUIS ENRIQUE MASCOTE CANO

Chihuahua, Chih., 30 de noviembre de 2025

Introducción

La extracción de conocimiento en grandes volúmenes de datos implica identificar patrones, estructuras internas y relaciones no evidentes. Dos herramientas fundamentales en este proceso son los algoritmos de agrupación (clustering) y los métodos de reducción de dimensionalidad.

El clustering permite organizar datos sin etiquetas en grupos que exhiben similitud interna, ayudando a descubrir segmentos, perfiles o comportamientos recurrentes. Por otro lado, la reducción de dimensionalidad simplifica datos de alta dimensión al preservar su estructura esencial, facilitando la visualización, el preprocesamiento y el rendimiento de algoritmos posteriores.

Este reporte presenta tres algoritmos de agrupación y dos de reducción de dimensionalidad, describiendo su funcionamiento, parámetros clave, ventajas, limitaciones y ejemplos. Finalmente, se incluye una comparativa práctica que orienta sobre cuándo utilizar cada técnica.

Algoritmos de agrupación

A continuación, se describen tres métodos fundamentales dentro del análisis no supervisado: K-means, clustering jerárquico aglomerativo y DBSCAN.

K-means

Principio de funcionamiento

K-means es un algoritmo basado en particiones. Su objetivo es dividir los datos en K clusters minimizando la variación interna. Opera mediante el siguiente proceso iterativo:

1. Elegir K centroides iniciales.
2. Asignar cada punto al centroide más cercano (distancia euclidiana).
3. Recalcular los centroides como el promedio de los puntos asignados.
4. Repetir pasos 2–3 hasta convergencia.

Parámetros clave

- K: número de clusters.
- Distancia: típicamente euclidiana.
- Inicialización: método k-means++ recomendado para estabilidad.

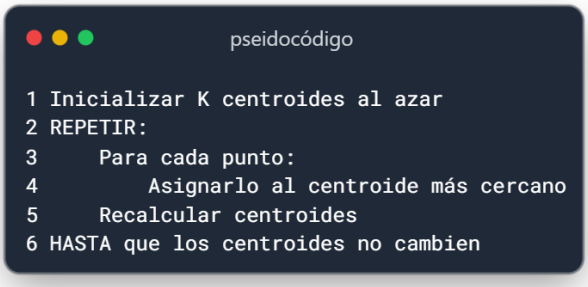
Ventajas

- Rápido y eficiente incluso con grandes datasets.
- Fácil de implementar.
- Produce clusters compactos y bien definidos.

Limitaciones

- Requiere definir K previamente.
- Sensible a outliers y valores iniciales.
- Solo detecta clusters esféricos.

Ejemplo en pseudocódigo



Clustering jerárquico aglomerativo

Principio de funcionamiento

Este método construye una jerarquía de clusters mediante un enfoque bottom-up:

1. Cada punto inicia como un cluster independiente.
2. Se fusionan los dos clusters más cercanos.
3. El proceso continúa hasta que todos los datos forman un solo cluster.

El resultado se presenta mediante un dendrograma, que permite visualizar la estructura de fusión.

Parámetros clave

- Linkage: single, complete, average, ward.
- Distancia: euclidiana o Manhattan.
- Umbral de corte: decide cuántos clusters finales se formarán.

Ventajas

- No requiere definir K desde el inicio.

- Produce una representación clara mediante dendrogramas.
- Puede capturar clusters no esféricos.

Limitaciones

- Escalabilidad limitada: complejidad (O(n^2)).
- Sensible a la elección del tipo de linkage.
- No re-asigna puntos después de una fusión.

Flujo de ejemplo

1. Crear un cluster por punto
2. Mientas existan más de 1 cluster:
   - Calcular distancias entre clusters
   - Fusionar los dos más cercanos
3. Generar dendrograma

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

Principio de funcionamiento

DBSCAN agrupa puntos basándose en densidad:

- Un punto es "central" si tiene suficientes vecinos dentro de una distancia.
- Un punto es "de borde" si está cerca de un central, pero no tiene suficientes vecinos.
- Puntos aislados se clasifican como ruido.

Forma clusters según regiones densas, permitiendo detectar estructuras complejas.

Parámetros clave

- ε (epsilon): radio máximo para considerar vecinos.
- minPts: número mínimo de vecinos para formar un punto central.

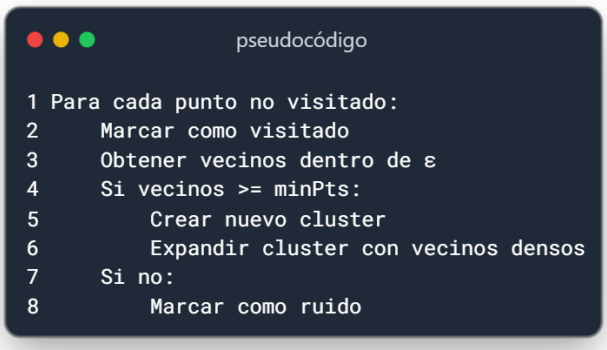
Ventajas

- Identifica clusters de forma arbitraria.
- Maneja outliers naturalmente.
- No requiere indicar el número de clusters.

Limitaciones

- Sensible a la escala de los datos.
- Difícil de ajustar en datos de densidades variadas.

Ejemplo de pseudocódigo



Algoritmos de reducción de dimensionalidad

Análisis de Componentes Principales (PCA)

Fundamento matemático

PCA proyecta los datos en nuevas variables llamadas componentes principales, que son combinaciones lineales ortogonales de las variables originales.

Se basa en la descomposición en valores propios de la matriz de covarianza:

1. Estandarizar datos.
2. Calcular matriz de covarianza.
3. Obtener eigenvalues y eigenvectors.
4. Ordenar componentes por varianza explicada.

Parámetros clave

- Número de componentes: cuánta varianza se desea conservar.
- Escalado previo: fundamental para evitar que variables grandes dominen.

Ventajas

- Reduce dimensionalidad preservando la varianza.
- Facilita visualización en 2D/3D.
- Mejora eficiencia computacional.

Limitaciones

- Asume relaciones lineales.
- Componentes difíciles de interpretar.

- Sensible a outliers.

Ejemplo ilustrativo

Dado X (n samples, d features):

- Estandarizar X
- Cov = $X^T X / (n-1)$
- Descomponer Cov en eigenvectors
- Proyectar X en los primeros k eigenvectors

t-SNE (t-Distributed Stochastic Neighbor Embedding)

Fundamento conceptual

t-SNE es un método no lineal diseñado para visualizar datos de alta dimensión. Modela las distancias entre puntos como probabilidades, preservando relaciones locales:

1. En alta dimensión: medir similitudes como distribuciones gaussianas.
2. En baja dimensión: usar distribuciones t de Student para evitar "colapso".
3. Minimizar la divergencia KL entre ambas distribuciones.

Parámetros clave

- Perplexity: regula el tamaño del vecindario local.
- Learning rate: controla velocidad de convergencia.
- Número de iteraciones.

Ventajas

- Excelente para visualizar clusters.
- Captura relaciones complejas no lineales.
- Produce gráficos intuitivos.

Limitaciones

- Costoso computacionalmente.
- No preserva distancias globales.
- No es adecuado para entrenamiento posterior del modelo.

Ejemplo conceptual

- Calcular similitud entre puntos en alta dimensión
- Inicializar posiciones 2D

- Ajustar posiciones minimizando KL divergence
- Mostrar puntos proyectados

Comparativa y conclusiones

Cuando usar clustering vs reducción de dimensionalidad

Objetivo	Clustering	Reducción de dimensionalidad
Agrupar datos sin etiquetas	✓	✗
Visualizar datos complejos	✗	✓
Detectar estructuras internas	✓	✓
Preprocesar para otro modelo	Poco usado	Muy usado
Manejo de ruido	DBSCAN excelente	PCA puede amplificar ruido

Situaciones prácticas

- Usar clustering cuando se requiere segmentar clientes, detectar patrones sin etiquetas o identificar regiones densas (ej. DBSCAN para detectar anomalías en sensores).
- Usar PCA o t-SNE cuando el dataset tiene muchas variables y se desea visualización o mejorar rendimiento de modelos posteriores.
- Usar ambos cuando se desea visualizar clusters obtenidos por K-means en un espacio reducido por PCA.

Conclusiones

Los algoritmos de clustering y reducción de dimensionalidad son pilares fundamentales del análisis no supervisado. K-means, el clustering jerárquico y DBSCAN permiten descubrir estructuras internas con diferentes enfoques: particiones, jerarquías y densidad. Por otro lado, PCA y t-SNE facilitan comprender conjuntos complejos mediante proyecciones lineales y no lineales.

Su correcta elección depende del problema: mientras que el clustering enfoca grupos, la reducción de dimensionalidad facilita visualización, compresión y preprocesamiento. En conjunto, ambas herramientas permiten extraer conocimiento valioso de datasets de alta complejidad.

Referencias

IBM. (2023). *What is clustering?* IBM Cloud Learn Hub. https://www.ibm.com/topics/clustering

IBM. (2023). *What is dimensionality reduction?* IBM Cloud Learn Hub. https://www.ibm.com/topics/dimensionality-reduction

Scikit-learn. (2024). *Clustering: K-means, hierarchical, DBSCAN*. https://scikit-learn.org/stable/modules/clustering.html

Scikit-learn. (2024). *Decomposition: PCA, Kernel PCA, NMF*. https://scikit-learn.org/stable/modules/decomposition.html

Google Developers. (2024). *Clustering methods overview*. https://developers.google.com/machine-learning/clustering

Towards Data Science. (2023). *Understanding t-SNE for high-dimensional data visualization*. https://towardsdatascience.com/understanding-t-sne-5a3c5862075d

Analytics Vidhya. (2024). *DBSCAN explained: Density-based clustering*. https://www.analyticsvidhya.com/blog/2021/06/dbscan-clustering-explained/