Universidad Tecnológica de Chihuahua
Tecnologías de la Información

Algoritmos de agrupación

Alumno:

Jatzel Israel Cruz Castruita

Grupo:

IDGS91N

Materia:

Extracción de Conocimiento en Bases de Datos

Docente:

Enrique Mascote

Introducción

El análisis de datos ha adquirido un papel fundamental en la comprensión y solución de problemas complejos en campos como la inteligencia artificial, la ciencia de datos, la medicina y los negocios. Entre las técnicas más utilizadas para explorar información sin etiquetas y para manejar grandes volúmenes de variables se encuentran los algoritmos de clustering y los métodos de reducción de dimensionalidad. El clustering permite identificar patrones naturales y agrupar elementos similares dentro de un conjunto de datos, facilitando la detección de estructuras internas y comportamientos comunes. Por otro lado, la reducción de dimensionalidad tiene como objetivo simplificar los datos eliminando ruido y redundancia, preservando al mismo tiempo la información esencial que describe el sistema.

Este documento presenta un estudio comparativo y detallado de ambos enfoques, destacando sus fundamentos, parámetros clave, ventajas, limitaciones y aplicaciones prácticas. Además, se abordan ejemplos ilustrativos que facilitan la comprensión de su funcionamiento. La finalidad es proporcionar una visión integral que permita reconocer en qué situaciones conviene emplear cada método y cómo pueden complementarse para obtener análisis más completos y precisos.

Algoritmos de agrupación

K-means

Principio de funcionamiento

K-means es un algoritmo de agrupación basado en centroides.
Divide los datos en K grupos, donde cada grupo tiene un centro llamado centroide.

El proceso es iterativo:

1. Se eligen K centroides iniciales (aleatorios o usando métodos como k-means++).
2. Cada punto se asigna al centroide más cercano.
3. Los centroides se recalculan como el promedio de los puntos asignados.
4. Se repiten los pasos 2–3 hasta que los centroides ya no cambian casi nada.

Parámetros clave

K

Número de clústeres que quieres formar (es obligatorio definirlo).

Máximo de iteraciones

Limita cuánto puede repetir el algoritmo.

Criterio de convergencia

Qué tan pequeño debe ser el cambio en los centroides para detener el algoritmo.

Ventajas y limitaciones

Ventajas

- Muy rápido y simple de implementar.
- Funciona bien cuando los clústeres son esféricos y tienen tamaños similares.
- Escala bien a grandes cantidades de datos.

Limitaciones

- Debes elegir K, no se detecta automáticamente.
- Sensible a valores atípicos (outliers).
- No funciona bien si los clústeres:
- no tienen forma esférica,
- tienen densidades muy distintas,
- están muy cerca unos de otros.
- Depende de la selección inicial de centroides.

Pseudocódigo

```
Elegir K número de clústeres
Inicializar K centroides

Repetir hasta convergencia:
    Para cada punto P:
        Asignar P al centroide más cercano

    Para cada clúster:
        Recalcular su centroide tomando el promedio de los puntos asignados

Fin
```

Algoritmos de reducción de dimensionalidad

Análisis de Componentes Principales (PCA)

Fundamento matemático o conceptual

PCA es un método de reducción de dimensionalidad lineal que busca representar los datos usando menos variables, pero manteniendo la mayor cantidad posible de información (varianza).

1. Su idea central:
2. Los datos tienen direcciones donde varían más que en otras.
3. PCA identifica esas direcciones principales.
4. Para encontrarlas:
    - Calcula la matriz de covarianza de los datos.
    - Obtiene sus autovectores (direcciones) y autovalores (qué tanta variación hay en esa dirección).
    - Ordena estas direcciones de mayor a menor varianza.

Parámetros clave

- n_components Cuántos componentes principales quieres conservar.
- Puede ser:
    - Un número entero (ej: 2 componentes)
    - Un porcentaje de varianza (ej: conservar el 95%)
- svd_solver (opcional) Indica el método para calcular PCA (full, randomized, auto).
- whiten (opcional) Hace que los componentes tengan varianza 1. Útil cuando se van a usar como entrada de modelos de ML.

Ventajas

- Muy rápido y eficiente incluso con miles de variables.
- Reduce ruido y correlación.
- Facilita visualizaciones en 2D/3D.
- Reduce riesgo de sobreajuste en modelos.
- Fácil de interpretar matemáticamente.

Limitaciones

- Solo captura relaciones lineales.
- Sensible a la escala de los datos (si no normalizas, la variable más grande domina).
- Los componentes no siempre son fáciles de interpretar (combinaciones de muchas variables).
- No funciona bien si la estructura es altamente no lineal (para eso se usa t-SNE, UMAP, autoencoders, etc.)

Pseudocódigo

```
Entrada: Matriz X (n muestras × d variables)

1. Normalizar los datos:
       Para cada columna:
            restar la media
            dividir entre la desviación estándar

2. Calcular la matriz de covarianza:
       C = cov(X)

3. Encontrar autovalores y autovectores de C

4. Ordenar los autovectores según los autovalores
       (de mayor varianza a menor)

5. Elegir los primeros k autovectores →
       formación de la matriz W_k

6. Proyectar los datos:
       X_reducido = X · W_k

Salida: X_reducido (n × k)
```

Enfoque	¿Para qué sirve?
Clustering	Encontrar grupos o patrones naturales en los datos. Clasificar sin etiquetas.
Reducción de dimensionalidad	Simplificar los datos eliminando variables redundantes, conservar estructura esencial y facilitar análisis, visualización o velocidad de modelos.

¿Cuándo usar cada uno?

La reducción de dimensionalidad se utiliza cuando el objetivo principal es simplificar, visualizar o preparar los datos antes de aplicar modelos; por ejemplo, cuando existen muchas variables correlacionadas o ruido. En cambio, el clustering se usa cuando se quiere descubrir grupos o patrones ocultos dentro de los datos. Ambos pueden complementarse: es común reducir la dimensionalidad primero (con PCA o autoencoders) y luego aplicar clustering para obtener grupos más nítidos. Sin embargo, si la meta final es solo encontrar clústeres, se prioriza el clustering; si la meta es explorar relaciones o reducir complejidad, se prioriza la reducción de dimensionalidad.

Situaciones prácticas donde uno tiene prioridad sobre el otro

Casos donde PRIORIDAD = Reducción de dimensionalidad

- Visualización de datos en 2D/3D (t-SNE, PCA).
- Acelerar modelos de ML que tienen demasiadas variables.
- Eliminar ruido o variables duplicadas.
- Preparar datos antes de aplicar clustering, regresión o clasificación.
- Cuando se quiere entender qué variables explican más variación.

Casos donde PRIORIDAD = Clustering

- Segmentar clientes en marketing.
- Detectar grupos o patrones en datos sin etiquetas.
- Identificar anomalías o puntos aislados (DBSCAN).
- Agrupación de imágenes similares.
- Clasificación preliminar cuando no se tienen categorías definidas.

Conclusión

En lo personal, después de comparar el clustering con los métodos de reducción de dimensionalidad, me queda claro que ambos son herramientas que se complementan más de lo que compiten. Para mí, el clustering es ideal cuando realmente quiero descubrir patrones ocultos y entender cómo se agrupan los datos de manera natural, mientras que la reducción de dimensionalidad la veo como una forma de “limpiar” y simplificar el panorama antes de tomar decisiones. Siento que, al usarlos juntos, puedo obtener una visión mucho más completa: primero reduzco el ruido y la complejidad, y luego identifico grupos más definidos. En mi experiencia, combinar ambos enfoques no solo hace el análisis más claro, sino que también me permite interpretar mejor los resultados y tener más confianza en las conclusiones que obtengo de los datos.

Referencias

Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: Data mining, inference, and prediction (2nd ed.). Springer.

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9(11), 2579–2605.

Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2065), 1–16.