UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

Tecnologías de la Información: Desarrollo y Gestión de Software

IV. Algoritmos de agrupación

IDGS91N - Kevin Iván Aguirre Silva
Extracción de Conocimiento en Bases de Datos - Ing. Luis Enrique Mascote Cano

Chihuahua, Chih., 29 de noviembre de 2025

Índice

1. Introducción

El presente documento aborda el estudio de los Algoritmos de Agrupación (Clustering) y los Algoritmos de Reducción de Dimensionalidad , pilares fundamentales en el campo de la Extracción de Conocimiento en Bases de Datos. Los métodos de clustering, como K-Means y DBSCAN , se centran en el aprendizaje no supervisado para descubrir estructuras y patrones inherentes agrupando datos similares. Por otro lado, las técnicas de reducción dimensional, como el Análisis de Componentes Principales (PCA) y t-SNE , buscan transformar los datos a un espacio de menor dimensión, conservando la información esencial, con el fin de mejorar la eficiencia y facilitar la visualización de conjuntos de datos complejos.

2. Algoritmos de agrupación

2.1. K-means

Principio de funcionamiento

El algoritmo K-Means es un método de aprendizaje no supervisado que agrupa datos en k clústers (grupos) buscando minimizar la distancia dentro de cada grupo a su centroide, que es el promedio de los puntos en ese clúster. Empieza con k centroides iniciales seleccionados aleatoriamente, asigna cada dato al clúster más cercano y recalcula los centroides de cada grupo. Este proceso se repite iterativamente hasta que los centroides dejan de cambiar o se alcanza un número máximo de iteraciones.

Parámetros clave

- **k:** Número de clústers a crear.
- **Centroides iniciales:** puntos seleccionados aleatoriamente o mediante técnicas específicas que sirven como centros iniciales.
- **Criterio de parada:** cuando los centroides dejan de cambiar o después de un número máximo de iteraciones.

Ventajas y limitaciones

Ventajas:

- Simple y fácil de entender e implementar.
- Computacionalmente eficiente para grandes conjuntos de datos.
- Bueno para encontrar grupos esféricos y bien separados.

Limitaciones:

- Requiere especificar k, el número de clústers, a priori.
- Sensible a la elección de los centroides iniciales (puede quedar atrapado en óptimos locales).
- No funciona bien con grupos de forma no esférica o tamaños muy diferentes.
- Sensible a puntos atípicos.

Ejemplo de aplicación simple

```
Inicializar k centroides aleatorios
Repetir hasta convergencia o máximo de iteraciones:
    Para cada punto:
        Asignar el punto al clúster cuyo centroide esté más cercano
    Para cada clúster:
        Recalcular el centroide como la media de puntos asignados
Fin
```

Este representa la base operativa del K-Means para agrupar datos en función de la cercanía a centroides iterativamente actualizados (Ramírez, 2024).

2.2. Clustering jerárquico (aglomerativo)

Principio de funcionamiento

El clustering jerárquico aglomerativo es un método de agrupación no supervisado que comienza considerando cada punto de datos como un clúster individual. Iterativamente, fusiona los dos clústeres más cercanos utilizando una medida de similitud o distancia, construyendo una jerarquía de clústeres anidados hasta que finalmente todos los puntos se agrupan en un único clúster o se cumple un criterio de parada.

Parámetros clave

- **Medida de distancia o similitud:** cómo se calcula la proximidad entre puntos o clústeres (p.ej., distancia euclidiana).
- **Método de enlace:** criterio para determinar la distancia entre clústeres (enlace simple, enlace completo, enlace promedio).
- **Criterio de parada:** número deseado de clústeres o un umbral de distancia para detener la fusión.

Ventajas y limitaciones

Ventajas:

- No requiere especificar el número de clústeres a priori.
- Produce una representación jerárquica que puede ser visualizada mediante dendrogramas.
- Útil para datos con estructuras jerárquicas naturales.

Limitaciones:

- Costoso computacionalmente para datasets grandes (alta complejidad).
- Sensible a ruido y datos atípicos.
- Resultados pueden depender mucho de la medida de distancia y método de enlace usados.

Ejemplo de aplicación simple

```
Inicializar cada punto como un clúster individual
Calcular matriz de distancias entre todos los clústeres
Repetir hasta que quede un solo clúster o criterio de parada:
    Encontrar los dos clústeres más cercanos
    Fusionar estos dos clústeres en uno solo
    Actualizar la matriz de distancias con el nuevo clúster
Fin
```

Este algoritmo construye la jerarquía de clústeres mediante fusiones sucesivas basadas en medidas de proximidad, facilitando análisis y segmentaciones jerárquicas de los datos (Noble, 2025).

2.3. DBSCAN

Principio de funcionamiento

El algoritmo DBSCAN (Density-Based Spatial Clustering of Applications with Noise) agrupa puntos en clústeres basándose en la densidad espacial, identificando regiones densas separadas por regiones de baja densidad. Clasifica los puntos en tres tipos: puntos núcleo (core) que tienen al menos un número mínimo de vecinos dentro de un radio específico, puntos frontera o alcanzables que están cerca de un punto núcleo, y puntos ruido que no cumplen las condiciones anteriores y no pertenecen a ningún clúster.

Parámetros clave

- (epsilon): radio que define el vecindario alrededor de un punto.
- MinPts: número mínimo de puntos que deben estar presentes dentro del radio ε para considerar un punto como núcleo.

Ventajas y limitaciones

Ventajas:

- Puede identificar clústeres de forma arbitraria, no solo esférica.
- Identifica automáticamente puntos ruido (outliers).
- No requiere especificar el número de clústeres a priori.

Limitaciones:

- Sensible a la selección de parámetros ε y MinPts.
- No funciona bien con densidades variadas, ya que un ε fijo puede no ser óptimo para todos los clústeres.
- Puede ser costoso en cómputo para grandes volúmenes de datos si no se optimiza.

Ejemplo de aplicación simple

```
Para cada punto no visitado:
    Marcar punto como visitado
    Encontrar puntos vecinos dentro del radio ε
    Si el número de vecinos >= MinPts:
        Crear nuevo clúster y añadir el punto
        Para cada vecino:
            Si no está visitado:
                Marcar como visitado
                Encontrar sus vecinos y ampliar clúster si cumple MinPts
            Si no pertenece aún a ningún clúster:
                Añadir al clúster actual
    Si no cumple MinPts:
        Marcar punto como ruido
Fin
```

DBSCAN es efectivo para datos con ruido y para descubrir clusters con formas complejas, diferenciándose así de métodos como K-Means (Fernández, s.f.).

2.4. Gaussian Mixture Models

Principio de funcionamiento

Gaussian Mixture Models (GMM) es un modelo probabilístico que asume que los datos se generan a partir de una mezcla de múltiples distribuciones gaussianas, cada una con su propia media y covarianza. A diferencia de K-Means, GMM permite que un punto tenga probabilidades de pertenecer a varios clústeres, ajustando los parámetros de las gaussianas mediante el algoritmo de Expectation-Maximization (EM), que alterna entre estimar las probabilidades de pertenencia y maximizar los parámetros de las distribuciones hasta convergencia.

Parámetros clave

Número de componentes gaussianas (número de clústeres).

Parámetros iniciales de medias, covarianzas y pesos de cada gaussiana.

Criterio de convergencia para el algoritmo EM.

Ventajas y limitaciones

Ventajas:

Puede modelar clústeres con formas elípticas y tamaños desiguales.

Asigna probabilidades de pertenencia, lo que refleja incertidumbre y solapamiento en clústeres.

Más flexible que K-Means para datos complejos.

Limitaciones:

Puede ser computacionalmente costoso en grandes datos.

Requiere especificar el número de componentes a priori.

Sensible a la inicialización y puede converger a óptimos locales.

Ejemplo de aplicación simple

```
Inicializar parámetros de medias, covarianzas y pesos para cada componente
gaussiana
Repetir hasta convergencia:
    Paso Expectation (E):
        Calcular probabilidades de pertenencia de cada punto a cada gaussiana
    Paso Maximization (M):
        Actualizar medias, covarianzas y pesos usando las probabilidades
calculadas
Fin
Asignar puntos a clúster según mayor probabilidad de pertenencia
```

GMM es útil para clustering probabilístico y modelado de densidad, siendo adecuado para datos con distribución compleja y cuando se requiere una asignación suave a clústeres (Richaud, 2024).

3. Algoritmos de reducción de dimensionalidad

3.1. Análisis de Componentes Principales (PCA)

Fundamento matemático o conceptual

El Análisis de Componentes Principales (PCA) es una técnica de reducción de dimensionalidad que transforma un conjunto de variables posiblemente correlacionadas en un conjunto menor de variables no correlacionadas llamadas componentes principales. Matemáticamente, PCA calcula los vectores propios

(eigenvectors) y valores propios (eigenvalues) de la matriz de covarianza o correlación de los datos, buscando las combinaciones lineales de variables originales que maximicen la varianza. El primer componente principal es la dirección con la mayor varianza, el segundo es ortogonal al primero y tiene la siguiente mayor varianza, y así sucesivamente.

Parámetros clave

- Número de componentes a elegir para mantener, que determina la nueva dimensión reducida.
- Datos deben estar generalmente estandarizados (media 0, varianza 1) para evitar que variables con escalas mayores dominen.
- Criterio de varianza explicada para decidir cuántos componentes conservar.

Ventajas y limitaciones

Ventajas:

- Reduce la dimensionalidad manteniendo la mayor parte de la varianza.
- Facilita visualización y análisis, eliminando redundancia en datos correlacionados.
- No requiere información previa sobre las relaciones entre variables.

Limitaciones:

- Solo captura relaciones lineales entre variables.
- Los componentes pueden ser difíciles de interpretar debido a ser combinaciones lineales.
- Sensible a la escala de las variables si no se estandarizan.

Ejemplo de aplicación simple

```
Estandarizar datos (media 0, varianza 1)
Calcular matriz de covarianza/correlación de datos
Calcular valores propios y vectores propios de la matriz
Ordenar los valores propios de mayor a menor
Seleccionar los primeros k vectores propios según la varianza explicada deseada
Proyectar los datos originales en el espacio definido por esos vectores propios
```

PCA es ampliamente usado para simplificar datos complejos, facilitar la visualización y mejorar la eficiencia de otros modelos mediante reducción dimensional (IBM, 2025).

3.2. t-SNE

Fundamento matemático o conceptual

t-SNE (t-distributed Stochastic Neighbor Embedding) es un algoritmo de reducción de dimensionalidad no lineal diseñado para visualizar datos de alta dimensión en espacios de menor dimensión, típicamente 2D o 3D. El algoritmo calcula primero una distribución de probabilidad sobre pares de puntos en el espacio original, donde puntos cercanos tienen alta probabilidad y puntos distantes baja probabilidad. Luego, en el espacio de menor dimensión, define una distribución similar usando una distribución t de Student para evitar el problema de la alineación de puntos distantes. Se optimizan las posiciones en el espacio reducido minimizando la divergencia de Kullback-Leibler entre ambas distribuciones mediante descenso de gradiente.

Parámetros clave

- Perplejidad: controla el número efectivo de vecinos usados para calcular las distribuciones de probabilidad; afecta la densidad de agrupamiento.
- Learning rate (tasa de aprendizaje): controla la velocidad de actualización en la optimización.
- Número de iteraciones: cantidad de pasos del descenso de gradiente para optimizar la representación.

Ventajas y limitaciones

Ventajas:

- Capta estructuras no lineales en los datos manteniendo relaciones locales.
- Muy efectivo para visualizar agrupaciones complejas en espacios de baja dimensión.
- Maneja bien datos con grupos variados y formas complejas.

Limitaciones:

- No es adecuado para reducción dimensional para modelos predictivos por perder interpretabilidad global.
- Computacionalmente costoso en conjuntos de datos muy grandes.
- Sensible a configuración de hiperparámetros como la perplejidad.

Ejemplo de aplicación simple

```
Calcular similitudes de pares en espacio alto dimensional con distribución
gaussiana
Inicializar aleatoriamente posiciones en espacio bajo dimensional
Repetir hasta convergencia o máximo número de iteraciones:
    Calcular similitudes de pares en espacio bajo dimensional con distribución t
de Student
    Calcular divergencia Kullback-Leibler entre las dos distribuciones
    Actualizar posiciones minimizando la divergencia usando descenso de gradiente
Fin
```

t-SNE es muy útil en exploración y visualización de datos de alta dimensión, especialmente en análisis de agrupamientos y detección de patrones (Datacamp, 2024).

3.3. Análisis Discriminante Lineal (LDA)

Fundamento matemático o conceptual

El Análisis Discriminante Lineal (LDA) es una técnica supervisada de reducción de dimensionalidad y clasificación que busca encontrar una combinación lineal de variables que maximice la separación entre múltiples clases. Matemáticamente, LDA encuentra un espacio de menor dimensión donde se maximiza la razón entre la varianza entre clases y la varianza dentro de las clases. Esto se logra mediante el cálculo de las matrices de dispersión dentro de clases (W) y entre clases (B), y la resolución del problema generalizado de valores propios para encontrar vectores que optimizan.

Parámetros clave

- Número de componentes discriminantes: máximo k – 1 dónde k es el número de clases.
- Supuestos de normalidad multivariante para cada clase con matrices de covarianzas iguales.
- Datos etiquetados para guiar la proyección hacia la mejor separación de clases.

Ventajas y limitaciones

Ventajas:

- Mejora la separación de clases en problemas supervisados.
- Eficiente computacionalmente y fácil de interpretar.
- Proporciona un clasificador lineal óptimo bajo supuestos gaussianos con covarianzas iguales.

Limitaciones:

- Requiere que las clases sigan distribuciones normales y tengan covarianzas similares.
- Limitado a fronteras de decisión lineales.
- No funciona bien cuando las clases no son linealmente separables o distribuidas de forma no gaussiana.

Ejemplo de aplicación simple

```
Calcular la media de cada clase y la media global
Calcular matrices de dispersión dentro de clases (W) y entre clases (B)
Resolver el problema de valores propios para \(W^{-1}B\)
Ordenar vectores propios según valores propios descendentes
Seleccionar los primeros k vectores para proyectar los datos
Proyectar datos en el nuevo espacio lineal para clasificación o visualización
```

LDA es una herramienta poderosa para clasificación supervisada y reducción de dimensionalidad cuando los datos cumplen sus supuestos estadísticos (cienciadedatos.net, 2016).

3.4. Autoencoders

Fundamento matemático o conceptual

Un Autoencoder es una red neuronal diseñada para aprender una representación comprimida (codificación) de los datos de entrada, con el objetivo de reconstruirlos lo más fielmente posible en la salida. Consta de dos partes: el encoder, que transforma los datos de alta dimensión en un espacio de menor dimensión (latente), y el decoder, que reconstruye los datos originales desde esta representación comprimida. El entrenamiento busca minimizar la función de pérdida, típicamente el

error cuadrático medio entre entrada y salida, ajustando los pesos de la red mediante retropropagación.

Parámetros clave

- Estructura y tamaño de la capa latente (dimensión reducida).
- Número de capas y neuronas en encoder y decoder.
- Función de activación en cada capa.
- Función de pérdida para comparar entrada y salida (por ejemplo, MSE).
- Parámetros de entrenamiento: tasa de aprendizaje, número de épocas, tamaño de lote (batch size).

Ventajas y limitaciones

Ventajas:

- Puede capturar representaciones no lineales complejas de los datos.
- Flexible en arquitectura y aplicable a distintos tipos de datos (imágenes, secuencias).
- Útil para reducción dimensional, detección de anomalías y generación de datos.

Limitaciones:

- Requiere cantidad significativa de datos para un buen entrenamiento.
- Puede sobreajustarse y perder generalización con arquitectura muy grande o pocos datos.
- La interpretación de la representación latente no siempre es clara.

Ejemplo de aplicación simple

```
Definir encoder con capas que reducen dimensión de entrada a latente
Definir decoder con capas que reconstruyen desde dimensión latente a original
Inicializar pesos de red neuronal
Para cada época:
    Para cada lote de datos:
        Codificar datos con encoder
        Decodificar codificación con decoder
        Calcular pérdida (diferencia entre entrada y salida)
        Propagar error y actualizar pesos (retropropagación)
Fin
```

Los autoencoders son herramientas poderosas para reducción no lineal de dimensionalidad y extracción automática de características, especialmente en contextos con grandes volúmenes de datos y relaciones complejas (EBIS Business Techschool, 2024).

4. Comparativa y conclusiones

4.1. Clustering vs. Reducción de Dimensionalidad

El Clustering (como K-Means, Jerárquico, DBSCAN y GMM) tiene como prioridad descubrir estructuras inherentes en los datos sin etiquetas, agrupando puntos similares para la segmentación o el descubrimiento de patrones no obvios.

- Prioridad del Clustering: Se utiliza cuando el objetivo es la segmentación de clientes, la detección de comunidades o el análisis exploratorio para entender la distribución natural de los datos.

Por otro lado, la Reducción de Dimensionalidad (como PCA, t-SNE, LDA y Autoencoders) tiene como objetivo transformar los datos a un espacio de menor dimensión. Esto se hace con el fin de mejorar la eficiencia computacional de modelos posteriores, facilitar la visualización o eliminar la redundancia y el ruido.

- Prioridad de la Reducción de Dimensionalidad: Se prefiere cuando los datos son demasiado grandes o tienen muchas características (el problema de la "maldición de la dimensionalidad") para mejorar la velocidad del entrenamiento de modelos supervisados (como PCA o LDA) o para crear visualizaciones 2D/3D comprensibles de estructuras de alta dimensión (como t-SNE).

4.2. Situación Práctica de Prioridad

Si el objetivo es encontrar los 5 perfiles principales de clientes de una tienda en línea sin saber cuáles son de antemano, el Clustering (por ejemplo, K-Means o GMM) toma prioridad. Si, en cambio, se tienen 1,000 características de producto y se quiere acelerar un modelo de predicción de precios o visualizar las relaciones complejas entre esas características, la Reducción de Dimensionalidad (por ejemplo, PCA o Autoencoders) es la herramienta principal. De hecho, es común que la reducción de dimensionalidad se use como pre-procesamiento antes de aplicar un algoritmo de clustering.

4.3. Conclusiones generales

Los algoritmos presentados representan herramientas fundamentales en el Aprendizaje No Supervisado (Clustering, PCA, t-SNE, Autoencoders) y Supervisado (LDA) para la Extracción de Conocimiento en Bases de Datos.

1. K-Means y Jerárquico son sencillos y eficientes, pero tienen limitaciones con formas de clústeres no esféricas.
2. DBSCAN y GMM ofrecen mayor flexibilidad para formas complejas y asignaciones probabilísticas, respectivamente, pero son más sensibles a la elección de parámetros o más costosos computacionalmente.
3. PCA es el método lineal estándar para la reducción de dimensionalidad, maximizando la varianza.
4. t-SNE es sobresaliente para la visualización de estructuras complejas, priorizando las relaciones locales.
5. LDA es único al ser un método supervisado que optimiza la separación de clases.
6. Los Autoencoders ofrecen una poderosa opción no lineal para la reducción dimensional mediante redes neuronales.
7. La selección del algoritmo más adecuado siempre debe basarse en la estructura subyacente de los datos y los requisitos específicos del proyecto

(rapidez, interpretabilidad, capacidad de manejar ruido, formas de clústeres, o la necesidad de etiquetas).

5. Referencias

cienciadedatos.net. (Septiembre de 2016). *Análisis discriminante lineal (LDA) y análisis discriminante cuadrático (QDA)*. Obtenido de cienciadedatos.net: https://cienciadedatos.net/documentos/28_linear_discriminant_analysis_lda_y_quadratic_discriminant_analysis_qda

Datacamp. (21 de Agosto de 2024). *Introducción al t-SNE*. Obtenido de Datacamp: https://www.datacamp.com/es/tutorial/introduction-t-sne

EBIS Business Techschool. (15 de Noviembre de 2024). *Autoencoder: Guía Completa 2025*. Obtenido de EBIS Business Techschool: https://www.ebiseducation.com/autoencoder-guia-completa

Fernández, A. (s.f.). *DBSCAN en Python: aprende cómo funciona*. Obtenido de Ander Fernández Jauregui - Machine Learning & Software Development: https://anderfernandez.com/blog/dbscan-python/

IBM. (1 de Abril de 2025). *¿Qué es el análisis de componentes principales (PCA)?* Obtenido de IBM: https://www.ibm.com/mx-es/think/topics/principal-component-analysis

Noble, J. (13 de Febrero de 2025). *¿Qué es el clustering jerárquico?* Obtenido de IBM: https://www.ibm.com/es-es/think/topics/hierarchical-clustering

Ramírez, L. (30 de Octubre de 2024). *Algoritmo k-means: ¿Qué es y cómo funciona?* Obtenido de IEBS Business School: https://www.iebschool.com/hub/algoritmo-k-means-que-es-y-como-funciona-big-data/

Richaud, A. (24 de Junio de 2024). *K-Means vs Gaussian Mixture Models: ¿cuál usar y para qué sirve cada uno?* Obtenido de https://antonio-richaud.com/blog/archivo/publicaciones/36-kmeans-vs-gmm.html