

UNIVERSIDAD TECNOLÓGICA DE CHIHUAHUA

Tecnologías de la información



9N Extracción de Conocimiento en Bases de Datos

IV.1. Algoritmos de agrupación (25%)

Docente:

ING. LUIS ENRIQUE MASCOTE CANO

Alumno:

Jorge Alfonso Lopez Bustamante

IDGS 91N

martes, 2 de diciembre de 2025

Índice

Introducción	3
Algoritmos de agrupación	3
Clustering jerárquico (aglomerativo)	3
DBSCAN.....	3
Gaussian Mixture Models (GMM)	4
Algoritmos de reducción de dimensionalidad	4
Análisis Discriminante Lineal (LDA)	4
Autoencoders (AE).....	4
Comparativa y conclusiones	5
Referencias.....	5

Introducción

El clustering (agrupación) y la reducción de dimensionalidad son pilares del aprendizaje no supervisado. La agrupación permite descubrir patrones y segmentos naturales en datos sin etiquetas, mientras que la reducción de dimensionalidad transforma los datos a espacios más compactos conservando la información esencial, mitigando la “maldición de la dimensionalidad” y facilitando la visualización y el desempeño de modelos posteriores.

Algoritmos de agrupación

Clustering jerárquico (aglomerativo)

Principio de funcionamiento: parte de cada punto como un clúster individual y va fusionando los clústeres más cercanos iterativamente hasta formar una jerarquía completa representada en un dendrograma. La distancia entre clústeres se define mediante un criterio de enlace: single, complete, average o Ward (minimiza la varianza intra-clúster).

Parámetros clave: métrica de distancia (euclíadiana, Manhattan, etc.), criterio de enlace y número de clústeres (definido cortando el dendrograma).

Ventajas: interpretable, no requiere fijar k de inicio, útil en conjuntos pequeños/medianos, permite ver relaciones jerárquicas.

Limitaciones: costo computacional elevado en grandes datasets; sensible a la elección de enlace/métrica; no reubica puntos una vez fusionados.

Ejemplo (pseudocódigo):

- 1) Inicializa $C = \{ \{x_1\}, \{x_2\}, \dots, \{x_n\} \}$
- 2) Calcula matriz de proximidad D entre clústeres
- 3) Mientras $|C| > k$: fusiona los dos clústeres con menor distancia según el enlace; actualiza D
- 4) Corta el dendrograma a la altura deseada para obtener k clústeres.

DBSCAN

Principio de funcionamiento: define clústeres como regiones de alta densidad. Un punto núcleo tiene al menos MinPts vecinos dentro de un radio ϵ ; los puntos alcanzables desde núcleos forman clústeres; puntos con pocos vecinos son frontera y los restantes se consideran ruido.

Parámetros clave: ϵ (radio) y MinPts (mínimo de vecinos); métrica de distancia.

Ventajas: detecta formas arbitrarias, no necesita predefinir el número de clústeres, maneja ruido/outliers.

Limitaciones: elección sensible de ϵ y MinPts ; desempeño pobre con densidades muy dispares; resultados pueden variar por el orden de exploración.

Ejemplo (pseudocódigo):

Para cada punto p no visitado:

- Marca p como visitado y encuentra $N_\epsilon(p)$
- Si $|N_\epsilon(p)| \geq \text{MinPts}$: crea nuevo clúster y expándelo agregando puntos densidad-

alcanzables

- Si no: marca p como ruido (puede ser frontera posteriormente).

Gaussian Mixture Models (GMM)

Principio de funcionamiento: modela los datos como mezcla de M gaussianas multivariadas; asigna pertenencias probabilísticas a cada componente. Se entrena con Expectation–Maximization (EM): en la fase E calcula responsabilidades para cada punto/componente; en la fase M actualiza medias, covarianzas y pesos para maximizar la verosimilitud.

Parámetros clave: número de componentes M; tipo de covarianza (completa, diagonal, esférica); criterios de convergencia e inicialización.

Ventajas: captura clústeres elípticos y superpuestos; produce probabilidades de pertenencia; mayor flexibilidad que K-means.

Limitaciones: requiere M; sensible a inicialización y a mínimos locales; costo crece con dimensionalidad; asume componente gaussiana.

Ejemplo (pseudocódigo simplificado EM):

Inicializa $\{w_i, \mu_i, \Sigma_i\}$

Repite hasta converger:

$$E: r_{\{i\}}(x) \leftarrow w_i N(x|\mu_i, \Sigma_i) / \sum_j w_j N(x|\mu_j, \Sigma_j)$$

$$M: w_i \leftarrow (1/n) \sum_x r_i(x); \mu_i \leftarrow (\sum_x r_i(x) x) / \sum_x r_i(x); \Sigma_i \leftarrow (\sum_x r_i(x) (x - \mu_i)(x - \mu_i)^T) / \sum_x r_i(x)$$

Algoritmos de reducción de dimensionalidad

Análisis Discriminante Lineal (LDA)

Fundamento: técnica supervisada que busca una proyección lineal que maximice la separación entre clases (dispersión entre-clases) y minimice la dispersión intra-clase.

Para dos clases, la dirección óptima w maximiza $J(w) = (w^T S_b w) / (w^T S_w w)$, donde S_b es la matriz de dispersión entre-clases y S_w la intra-clase.

Parámetros clave: número de componentes (\leq número de clases – 1); regularización de S_w si es singular.

Ventajas: simple, interpretable, útil para preprocesamiento supervisado y clasificación; reduce a $(c-1)$ dimensiones.

Limitaciones: supone clases linealmente separables con covarianzas similares; sensible a outliers; menos eficaz con fronteras no lineales.

Ejemplo (pseudocódigo): calcula medias por clase y matrices S_w , S_b ; resuelve el problema de autovalores generalizado $S_b v = \lambda S_w v$; proyecta datos en las primeras direcciones v .

Autoencoders (AE)

Fundamento: redes neuronales encoder–decoder entrenadas para reconstruir la entrada; la capa “cuello de botella” aprende una representación de baja dimensión (no lineal). El entrenamiento suele incluir preentrenamiento y ajuste fino, y variantes como autoencoders denoising, esparsos o variacionales.

Parámetros clave: arquitectura (profundidad, neuronas), dimensión del código, función de pérdida (MSE, BCE), regularización, tasa de aprendizaje.

Ventajas: modela relaciones no lineales, permite mapeo bidireccional (codificación y decodificación), escalable a grandes datasets.

Limitaciones: requiere mayor cómputo y cuidado en hiperparámetros; puede sobreajustar; reproducibilidad depende de inicialización/semillas.

Ejemplo (pseudocódigo):

Define encoder f_{θ} : $x \rightarrow z$ y decoder g_{ϕ} : $z \rightarrow \hat{x}$; minimiza $L = \sum ||x - g_{\phi}(f_{\theta}(x))||^2$ mediante backprop; usa z como representación reducida.

Comparativa y conclusiones

Cuándo usar clustering vs. reducción de dimensionalidad:	
Clustering	Dimensionalidad
Cuando se busca segmentar sin etiquetas, descubrir patrones o anomalías; puede beneficiarse de reducir dimensiones previamente (p.ej., PCA/LDA/AE → clustering).	Cuando se requiere visualizar, comprimir o preparar datos para algoritmos supervisados/no supervisados; suele preceder a clustering o clasificación.

Situaciones prácticas: en datos de alta dimensión, aplicar reducción (LDA si hay etiquetas, AE si no lineal) y luego un algoritmo de agrupación (jerárquico/DBSCAN/GMM).

Conclusión: la elección depende de la estructura de los datos, disponibilidad de etiquetas y objetivos. Jerárquico ofrece interpretabilidad; DBSCAN detecta densidades irregulares; GMM aporta flexibilidad probabilística. LDA es adecuado para separaciones lineales supervisadas y autoencoders brindan representaciones no lineales potentes.

Referencias

- Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). *A density-based algorithm for discovering clusters in large spatial databases with noise*. KDD.
- Reynolds, D. A. (2009). *Gaussian Mixture Models*. Encyclopedia of Biometrics.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning* (2nd ed.). Springer.
- Fisher, R. A. (1936). *The use of multiple measurements in taxonomic problems*. Annals of Eugenics.
- Hinton, G. E., & Salakhutdinov, R. R. (2006). *Reducing the dimensionality of data with neural networks*. Science.