Procesamiento de Grandes Volúmenes de Datos

Conferencia 6: Paradigmas Avanzados de Clustering a Gran Escala

Deborah Famadas Rodríguez

Universidad de la Habana

20 de octubre de 2025

Escalando Métodos Basados en Densidad (DBSCAN)

El Cuello de Botella: Búsqueda de Vecindad Exacta

La operación central en DBSCAN es la `RegionQuery`, encontrar todos los puntos en un radio ϵ.

- Una implementación ingenua requiere $O(n^2)$ comparaciones de distancia.
- Las estructuras de indexación espacial (k-d trees) fallan en **alta dimensión**.

Solución Necesaria

Para escalar DBSCAN a N grande, necesitamos abandonar la búsqueda exacta de vecinos en favor de métodos aproximados.

Estrategias de Aproximación I: Particionamiento (Baja Dimensión)

Particionamiento Basado en Rejilla (Grid-Based)

- El espacio de datos se divide en una rejilla de celdas de tamaño ϵ.
- La búsqueda de vecinos para un punto se limita a su celda y las celdas adyacentes (máx $3^d - 1$ celdas).
- **Ventaja**: Reduce la complejidad a $O(n \log n)$ o incluso $O(n)$ en casos favorables.
- **Desafío**: Inviable para $d > 10$. El número de celdas crece exponencialmente.

Estrategias de Aproximación II: Alta Dimensión

Hashing Sensible a la Localidad (LSH)

- Técnica de **hash probabilístico** para la búsqueda de Vecinos Más Cercanos Aproximados (ANN).
- **Principio**: Puntos similares (cercanos) tienen una alta probabilidad de colisionar en el mismo cubo de hash.
- **Aplicación a DBSCAN**: La `RegionQuery` se reduce a buscar vecinos solo en los cubos relevantes.
- Conduce a un Approximate-DBSCAN con complejidad **sub-cuadrática**.

DBSCAN Distribuido con Particionamiento

Estrategia Básica: Particionar y Unir

- El dataset se divide en particiones no superpuestas $P_1, P_2, \ldots$.
- **Fase 1 (Local)**: Ejecutar DBSCAN localmente en cada partición P_i.
- **Fase 2 (Unión)**: Los clústeres locales se fusionan si son adyacentes.

El Problema del Borde

Los puntos en el borde de una partición pueden ser de núcleo (core points) o de frontera (border points) si se consideran los puntos de la partición adyacente. Esto requiere una fase de "halo.º "vecindad extendida.^entre particiones.

HDBSCAN: Evolución Jerárquica y Escalable

HDBSCAN: Clustering Jerárquico Basado en Densidad

- Elimina el parámetro ϵ fijo.
- Usa la **Distancia de Accesibilidad Mutua** y construye un Árbol de Esparcimiento Mínimo (MST).
- Aplica una técnica de poda para obtener la estructura jerárquica de densidad de los clústeres.

Escalamiento de HDBSCAN

El cuello de botella se traslada al cálculo del **MST**.

- El cálculo directo es $O(n^2)$.
- Se usan técnicas **ANN/LSH** para el cálculo aproximado de distancias, o métodos de particionamiento con límites (similar al DBSCAN distribuido).

Escalando el Paradigma Basado en Modelos (GMM)

Modelos de Mezcla Gaussiana (GMM)

- Asume que los datos provienen de una mezcla de distribuciones Gaussianas (clústeres elipsoidales con diferente varianza).
- Se entrena con el algoritmo **Expectation-Maximization (EM)**.
- **Cuello de Botella**: El paso M (Maximización) requiere el cálculo de estadísticas (medias, covarianzas) sobre **todo el dataset** (N) en cada iteración.

Solución para GMM: EM Estocástica (S-EM)

Algoritmo Stochastic EM (S-EM)

- **Principio**: En lugar de usar N puntos para actualizar los parámetros en el paso M, se usan **mini-lotes** muestreados aleatoriamente ($m \ll N$).
- Las actualizaciones se ponderan con una tasa de aprendizaje (learning rate), similar al Descenso de Gradiente Estocástico (SGD).

EM Tradicional (Batch)

- Precisión por iteración alta.
- Lento: $\mathcal{O}(N \cdot k \cdot d^2)$ por iteración.

S-EM (Estocástico)

- Precisión por iteración baja.
- Rápido: $\mathcal{O}(m \cdot k \cdot d^2)$ por iteración ($m \ll N$).

Patrones Arquitectónicos en Plataformas Distribuidas

Patrón 1: Iterativo-Sincronizado (Bulk Synchronous Parallel - BSP): Ej: K-Means paralelo

.

Patrón 2: Condensar-y-Refinar: Ej: BIRCH (mencionado previamente), Coresets

Patrón 3: Asíncrono (Parameter Server)

Idea: Se mantiene un servidor central que almacena los parámetros (ej. centroides, pesos de GMM). Los *workers* leen y escriben actualizaciones sin barreras de sincronización.

Validación y Evaluación a Gran Escala

Métricas de Validación Interna (Recordatorio)

- **Puntuación de Silueta**: Mide cuán similar es un punto a su clúster vs. otros clústeres. $\mathcal{O}(n^2)$.
- **Índice de Davies-Bouldin (DBI)**: Mide la dispersión intra-clúster relativa a la separación inter-clúster.

Desafío y Solución de Escalabilidad

El cálculo exacto de estas métricas es prohibitivo. Se requiere **estimación mediante muestreo estratégico** o cálculo distribuido de estadísticas **fusionables** (sumas, sumas cuadráticas).

Robustez II: Trimmed Clustering

Trimmed K-Means (TKM)

- **Idea**: Un algoritmo híbrido que combina la velocidad de K-Means con la solidez de DBSCAN.
- **Mecanismo**: Ignora una fracción β de los puntos que contribuyen con el mayor error cuadrático, tratándolos como ruido o outliers.
- El número de puntos a ignorar (β) es un parámetro de entrada.

Introducción al Clustering Justo (Fair Clustering)

Motivación Ética y de Gobernanza

Los algoritmos de clustering estándar pueden generar resultados sesgados, segregando o concentrando grupos demográficos específicos (basados en atributos protegidos: raza, género, edad).

- Ejemplo: Clústeres de crédito donde un grupo minoritario es sistemáticamente colocado en un clúster de alto riesgo.

Restricciones de Equidad: Equilibrio (Balance)

Nociones de Equidad

Equidad como Restricción: Se añade una restricción de equilibrio a la función de coste.

- **Min-Size Constraint**: Cada clúster C_j debe contener al menos un número mínimo m de puntos de cada grupo sensible G_i.
- **Proportional Balance**: La composición demográfica de cada clúster debe ser proporcional (o similar) a la composición demográfica del conjunto de datos completo.

¿Preguntas?