Procesamiento de Grandes Volúmenes de Datos

Conferencia 6: Clustering a Gran Escala

Deborah Famadas Rodríguez

Universidad de la Habana

14 de octubre de 2025

Aplicación 1: Segmentación de Usuarios y Entidades

Caso de Uso Principal

Agrupar usuarios según su comportamiento para:

- Marketing dirigido.
- Personalización de experiencias.
- Detección de anomalías.

Ejemplo Económico

Identificar concentraciones geográficas de industrias (clústeres industriales) para diseñar estrategias de desarrollo regional.

Aplicación 2: Deduplicación y Resolución de Entidades

Problema

En bases de datos masivas, es común encontrar registros duplicados o casi duplicados (perfiles de clientes, catálogos de productos).

Solución

El clustering a gran escala es un paso previo crucial para **identificar y consolidar** estas entidades.

Aplicación 3: Compresión Semántica

Técnica: Cuantificación Vectorial

El clustering permite representar un vasto conjunto de datos mediante un pequeño conjunto de prototipos (los centroides de los clústeres).

Aplicación

Es fundamental para la **compresión con pérdida** en datos de imagen, sonido y vídeo.

Aplicación 4: Descubrimiento Científico y Particionamiento

Descubrimiento Científico

Herramienta indispensable en:

- Análisis de expresión génica, identificar comunidades, agrupar documentos por tema.

Particionamiento Grueso

A menudo, el clustering es un paso intermedio para dividir un problema intratable en subproblemas manejables.

- *Ejemplo*: Particionar miles de millones de logs antes de aplicar modelos predictivos más complejos a cada partición.

Desafíos: Volumen y Dimensionalidad

Volumen (n masivo)

- Miles de millones de puntos de datos.
- Algoritmos con complejidad $O(n^2)$ no se utilizan.
- **Objetivo**: Complejidad lineal o casi lineal, $O(n)$.

Dimensionalidad (d elevado)

- **La Maldición de la Dimensionalidad**: Las métricas de distancia (ej. Euclidiana) pierden su contraste.
- Todos los puntos tienden a volverse equidistantes, socavando la noción de "proximidad".

Desafíos: Velocidad, Variedad y Veracidad

Velocidad (Datos en Streaming)

- Los datos llegan como un flujo continuo.
- Se deben procesar en **una sola pasada**.
- Invalida algoritmos iterativos que requieren múltiples pasadas.

Variedad y Veracidad

- **Variedad**: Datos de fuentes diversas (texto, imágenes, grafos). Requiere medidas de similitud flexibles.
- **Veracidad**: Datos ruidosos, con outliers y distribuciones sesgadas (skew).

Desafíos: Restricciones del Sistema

Limitaciones Prácticas

El diseño de algoritmos debe considerar:

- Memoria disponible por nodo.
- Ancho de banda de la red (coste de comunicación).
- Requisitos de latencia.

Interconexión de Desafíos

"¿qué tan bien encuentra los clústeres?.ª "¿qué tan útilmente particiona los datos para la tarea posterior?".

Sección 2: Paradigmas de Clustering

- **Particional**: Divide los datos en k grupos predefinidos.
- **Jerárquico**: Crea una jerarquía de clústeres en forma de árbol.
- **Basado en Densidad**: Define clústeres como regiones densas de puntos.
- **Basado en Grafos / Espectral**: Modela los datos como un grafo y busca particiones (cortes).

Paradigma 1: Clustering Particional (K-Means)

Idea Central

Encontrar una partición que **minimice una función objetivo global** (dispersión intra-clúster).

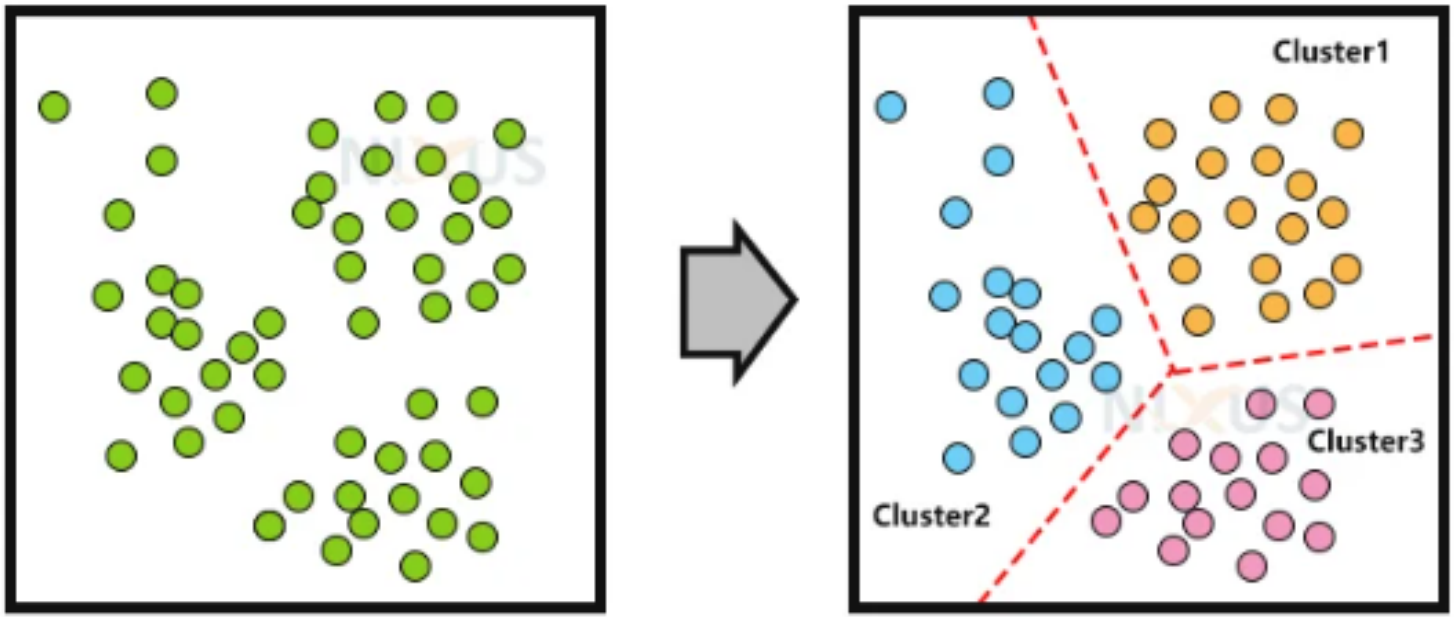
Fortalezas:

- Computacionalmente eficiente ($O(n)$).

Debilidades:

- Requiere pre-especificar k.

Clustering particional



Paradigma 2: Clustering Jerárquico

Idea Central

Agrupar objetos basados en su **conectividad y proximidad**, revelando agrupaciones a diferentes escalas en un dendrograma.

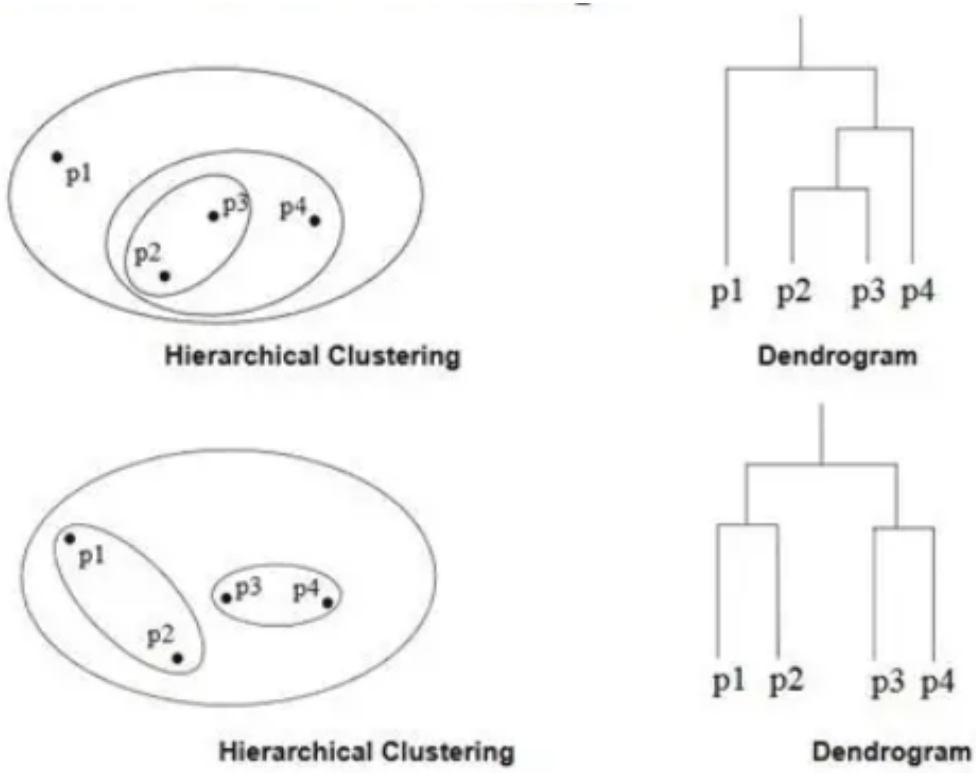
Fortalezas:

- No requiere pre-especificar k y el dendrograma proporciona visualización.

Debilidades:

- Computacionalmente costoso ($O(n^2 \log n)$ o $O(n^3)$).
- Inviable para n grande.
- Decisiones de fusión/división son irrevocables (codicioso).

Clustering jerarquico



Paradigma 3: Clustering Basado en Densidad (DBSCAN)

Idea Central

Define clústeres como **regiones densas de puntos**, separadas por regiones de baja densidad. Un clúster es un conjunto de puntos densamente conectados.

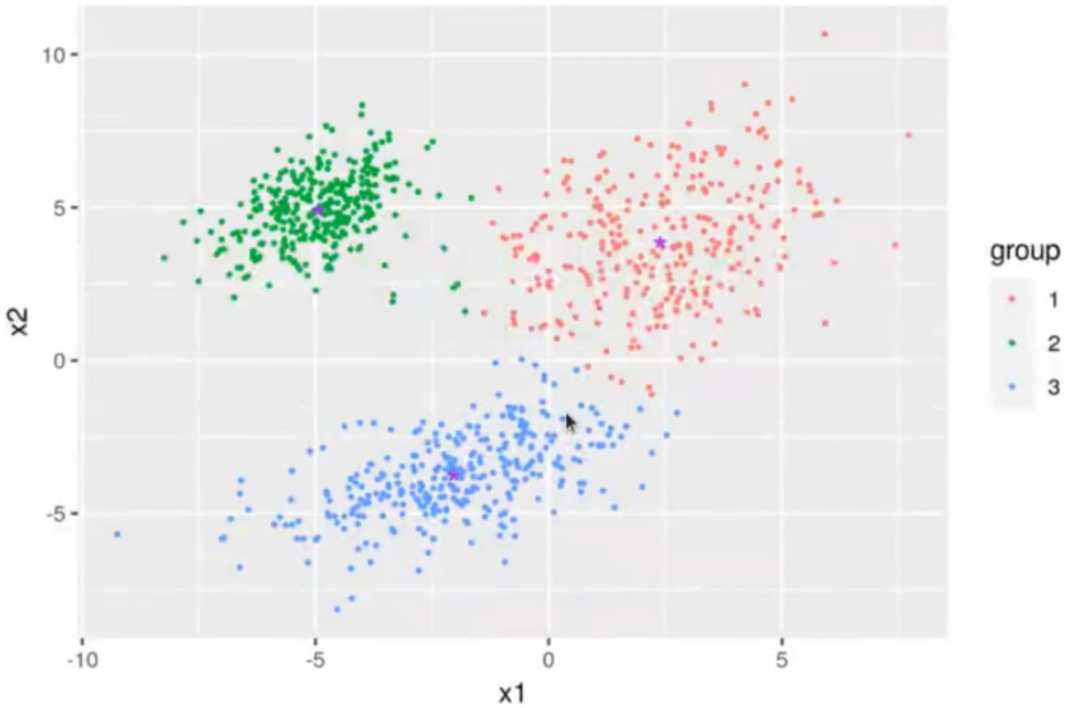
Fortalezas:

- Descubre clústeres de formas arbitrarias, robusto al ruido y a los outliers.

Debilidades:

- Sensible a los parámetros (*eps*, *MinPts*).
- Dificultad con clústeres de densidades variables.
- El rendimiento se degrada en alta dimensión.

Clustering basado en densidad



Paradigma 4: Clustering Basado en Grafos (Espectral)

Idea Central

El problema de clustering se reformula como un problema de **particionamiento de grafos**. El objetivo es encontrar ”cortes” que separen componentes débilmente conectados.

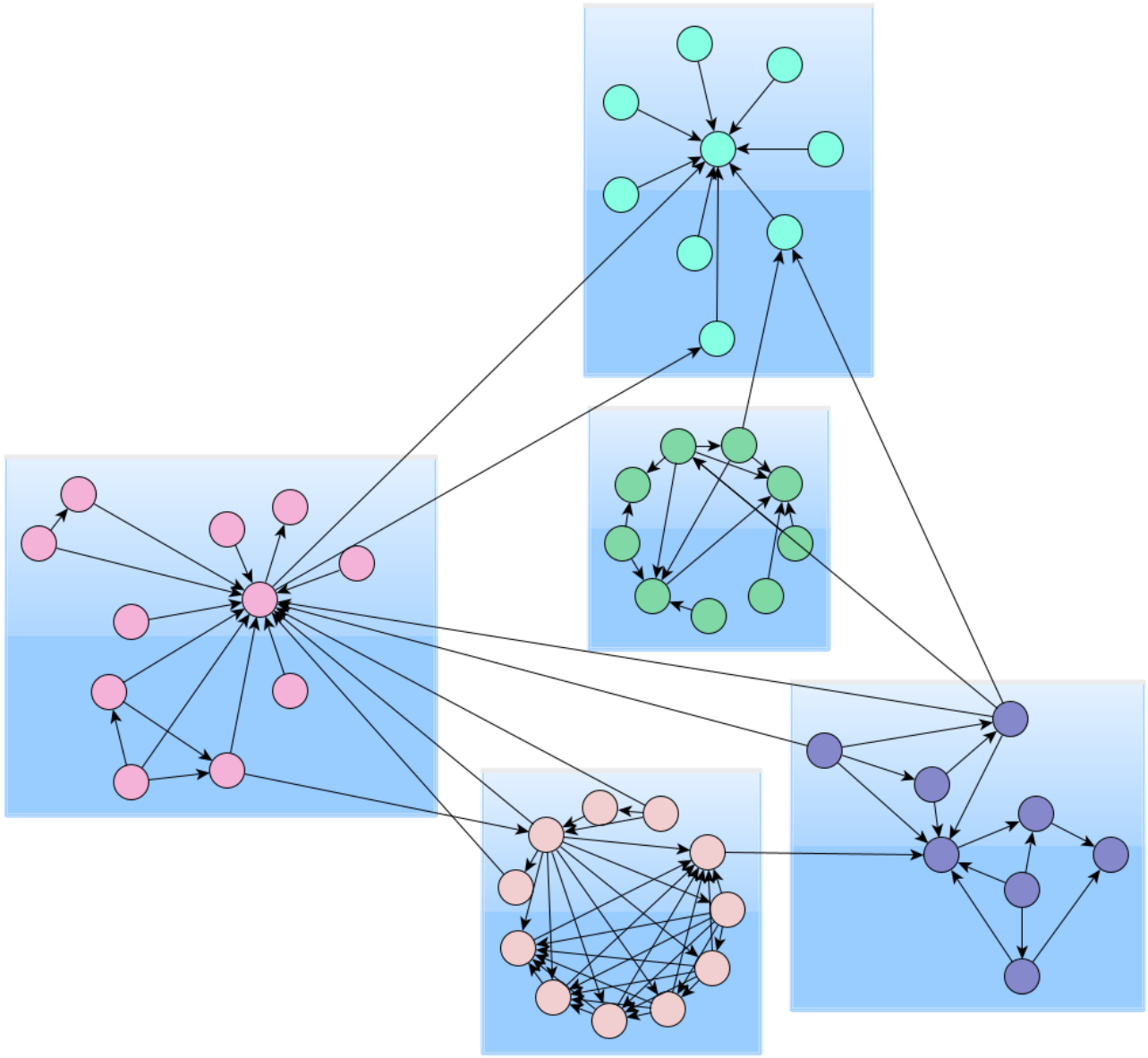
Fortalezas:

- Muy potente para descubrir estructuras complejas.

Debilidades:

- Construcción de la matriz de similitud requiere $O(n^2)$.
- Cálculo de autovectores es $O(n^3)$.
- Impracticable para n grande sin aproximaciones.

Clustering basado en grafos



Parte II: K-Means - De la Teoría a la Práctica Escalable

- A pesar de su simplicidad, k-means es un arquetipo fundamental.
- Es el punto de partida para comprender los desafíos del clustering a gran escala.

Función Objetivo: Minimizar la Suma de Errores Cuadráticos (SSE)

El objetivo es encontrar una partición $S = \{S_1, S_2, ..., S_k\}$ que minimice la inercia o varianza intra-clúster.

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

donde $\boldsymbol{\mu}_i$ es el centroide (media) de los puntos en el clúster S_i.

$$\boldsymbol{\mu}_i = \frac{1}{|S_i|} \sum_{\mathbf{x} \in S_i} \mathbf{x}$$

- Esta función codifica la intuición de que un buen clúster es aquel en el que los puntos están muy agrupados alrededor de su centro.

Algoritmo de Lloyd: Refinamiento Iterativo

Dado que encontrar la solución global es NP-difícil, se utiliza la heurística de Lloyd, que alterna dos pasos hasta la convergencia:

Paso de Asignación

Cada punto de datos se asigna al clúster cuyo centroide actual es el más cercano. (Particiona el espacio en un diagrama de Voronoi)

Paso de Actualización

Se recalcula el centroide de cada clúster como la media de todos los puntos asignados a él.

Importante: Converge a un mínimo **local**, no necesariamente al global.

Complejidad y la Maldición de la Dimensionalidad

Análisis de Complejidad

- Total: $O(n \cdot k \cdot d \cdot T)$, donde T es el número de iteraciones.

Impacto de la Maldición de la Dimensionalidad

En espacios de alta dimensión (d grande), la distancia euclidiana pierde significado. El contraste entre distancias intra-clúster e inter-clúster disminuye, degradando la calidad del clustering.

Mitigación

Preprocesamiento: Escalar características y, crucialmente, aplicar **reducción de dimensionalidad** (ej. PCA).

Sección 4: Marcos Teóricos para Escalar K-Means

Dos cuellos de botella principales:

1. **Sensibilidad a la inicialización**: Puede llevar a convergencia lenta y a óptimos locales de baja calidad.
2. **Procesamiento completo del dataset**: La necesidad de procesar todo el conjunto de datos en cada iteración.

Estrategias de Inicialización: K-Means++

Propone un método de siembra (seeding) más inteligente y probabilístico.

Algoritmo

1. El primer centroide se elige de manera uniforme y aleatoria.
2. Para cada centroide subsiguiente, cada punto de datos $\mathbf{x}$ se elige con una probabilidad proporcional a $D(\mathbf{x})^2$, donde $D(\mathbf{x})$ es la distancia al centroide más cercano ya seleccionado.

- **Intuición**: Favorece la selección de puntos que están lejos de los centroides existentes.
- **Garantía Teórica**: Se espera que la solución sea una aproximación $O(\log k)$ a la solución óptima.

Estrategias de Inicialización: K-Means‖

Problema con K-Means++

Su naturaleza secuencial (k pasadas sobre los datos) es un cuello de botella en entornos paralelos.

Solución de K-Means‖

Aborda esto mediante un **sobremuestreo (oversampling) en rondas paralelas**.

1. En lugar de seleccionar un solo punto, en cada ronda se muestrean $O(k)$ puntos en paralelo, usando la misma distribución de probabilidad $D(x)^2$.
2. Se realizan un pequeño número de rondas (ej. 5).
3. El conjunto de centroides candidatos resultante se agrupa (reclusteriza) para producir los k centroides iniciales.

Pizarra 2: La Intuición del Muestreo $D(x)^2$

- **Paso 1**: Se elige el primer centroide, c_1, al azar. Supongamos que cae en el Clúster 1.
- **Paso 2**: Se calcula $D(x)^2$ para todos los puntos.
  - Para puntos en el Clúster 1, $D(x)^2$ es pequeño.
  - Para puntos en otros clústeres (lejos de c_1), $D(x)^2$ es grande.
- **Paso 3**: La probabilidad de selección $P(x) \propto D(x)^2$ se concentra en los puntos más alejados.
- **Resultado**: Es muy probable que el siguiente centroide, c_2, se seleccione de un clúster aún no çubierto".

El muestreo $D(x)^2$ explota la estructura del problema: enfoca la atención en las regiones del espacio mal representadas por los centroides actuales.

Reducción de Datos: Mini-Batch K-Means

Concepto

Una aproximación estocástica del algoritmo de Lloyd.

- En lugar de utilizar todo el dataset, cada iteración utiliza una pequeña muestra aleatoria (**mini-batch**).
- Los centroides se actualizan basándose únicamente en la información de este mini-batch.
- **Compromiso**: Convergencia mucho más rápida a costa de una calidad de clúster potencialmente ligeramente inferior.
- **Ideal para**: Aprendizaje en línea (online) y datos que no caben en memoria.

Reducción de Datos: Coresets

Concepto

Un **coreset** es un pequeño subconjunto ponderado del dataset original que preserva la estructura del problema de k-means.

Garantía Formal

Para cualquier conjunto de k centroides, el coste SSE calculado en el coreset es una aproximación $(1 \pm \epsilon)$ del coste en el dataset completo.

Reducción de Datos: Coresets

Flujo de Trabajo

1. Construir un coreset (generalmente mucho más pequeño que los datos originales).
2. Ejecutar k-means (u otro algoritmo costoso) sobre el coreset.
3. La solución obtenida tiene garantías de ser una buena aproximación para el problema original.

Costes de Comunicación en K-Means Paralelo

Modelo Bulk Synchronous Parallel (BSP)

Los algoritmos se estructuran en "superpasos":

1. **Computación Local**: Cada nodo trabaja con sus datos.
2. **Comunicación Global**: Los procesadores intercambian datos.
3. **Sincronización de Barrera**: Se espera a que todos terminen la comunicación.

K-Means en BSP

El cuello de botella es la **comunicación**: la reducción (shuffle) de sumas parciales y la difusión (broadcast) de los nuevos centroides generan tráfico de red y latencia.

Tabla Comparativa: Técnicas de Escalamiento de K-Means

Técnica	Objetivo Principal	Garantía Aprox.	Rondas Com.	Adecuado para
k-means++	Mejor óptimo local, convergencia más rápida	$O(\log k)$	k (secuencial)	Datos estáticos, tamaño moderado
k-means$\|\|$	Inicialización paralela rápida y de alta calidad	$O(\log k)$	$O(\log \Psi)$ (paralelo)	Entornos distribuidos (MapReduce, Spark)
Mini-Batch	Baja memoria, aprendizaje en línea	No (heurístico)	N/A (por batch)	Streaming, datos que no caben en memoria
Coresets	Reducción de datos con garantía teórica	$(1 \pm \epsilon)$	1 (construcción)	Cuando se necesitan garantías teóricas fuertes

Parte III: Modelos de Clustering Avanzados a Escala

Más allá de k-means: escalando paradigmas para problemas del mundo real.

- Clústeres de formas no convexas.
- Ruido significativo.
- Datos en flujos continuos.

Escalar estos modelos avanzados requiere superar desafíos computacionales aún mayores.

Sección 5: Clustering en Entornos de Streaming

Restricciones del Streaming

- Los datos se procesan a medida que llegan.
- Típicamente en **una sola pasada**.
- Con memoria limitada.
- Prohíbe algoritmos que requieren acceso aleatorio a todo el dataset.

Estrategia Central: Micro-clústeres

No se almacenan los puntos individuales, sino un conjunto de **resúmenes estadísticos ligeros** llamados "micro-clústeres".

Micro-clúster y BIRCH

Micro-clúster / Clustering Feature (CF)

Un resumen de un grupo de puntos cercanos que almacena una tupla de estadísticas:

- N: Número de puntos.
- LS: Suma lineal de los puntos.
- SS: Suma de los cuadrados de los puntos.

La propiedad clave de los CF es que son **aditivos**.

Algoritmo BIRCH y el Árbol CF

BIRCH organiza los micro-clústeres (CFs) en una estructura de árbol balanceada (similar a un B-Tree) para inserciones y actualizaciones eficientes ($O(N)$).

Micro-clúster y BIRCH

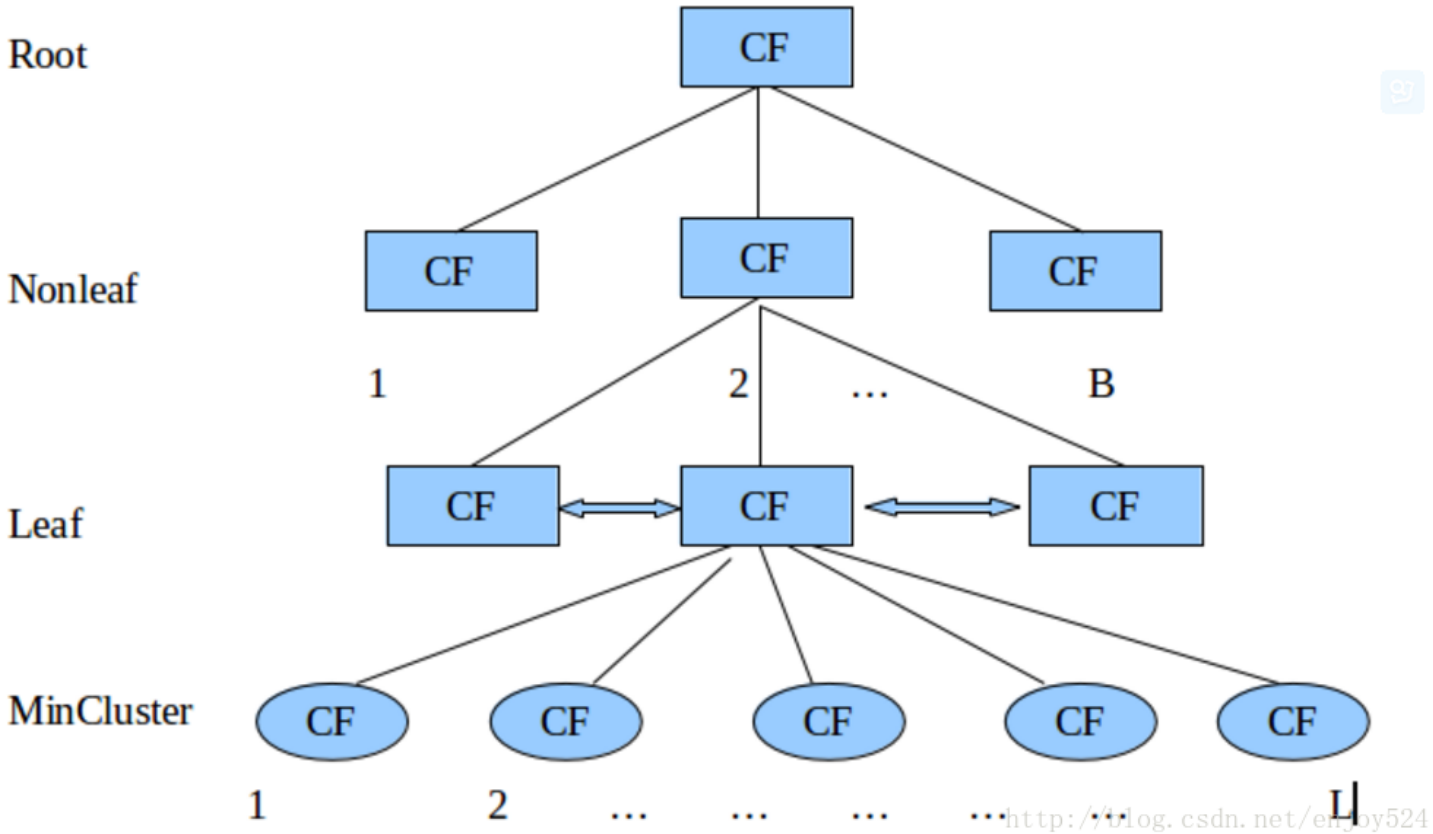


Figura: CF Tree

¿Preguntas?