

# UNIDAD 2: MINERÍA DE ELEMENTOS FRECUENTES

## ALGORITMOS DE PASADAS LIMITADAS

---

Blanca Vázquez y Gibran Fuentes-Pineda

7 de septiembre de 2021

- Si no se puede tener en memoria principal la base de datos y los contadores, se necesitan múltiples pasadas
- Muestreo
  - Muestreo aleatorio simple
  - Algoritmo Savasere, Omiecinski y Navathe
  - Algoritmo de Toivonen

## MUESTREO ALEATORIO SIMPLE (1)

- Se buscan los conjuntos frecuentes a partir de una muestra aleatoria de la base de datos
- Cada entrada de la base de datos se elige con una probabilidad  $p$ 
  - Si la base de datos es de tamaño  $n$ , tendríamos aproximadamente  $p \cdot n$  muestras
- Se aplica el algoritmo apriori, PCY, multietapa o multihash a la muestra

## MUESTREO ALEATORIO SIMPLE (2)

- Los conjuntos con soporte muy cercano al umbral pueden o no ser frecuentes en la muestra aleatoria
- Se pueden encontrar conjuntos frecuentes en la muestra que no lo son en la base de datos completa (falsos positivos)
  - Se pueden evitar verificando que sean frecuentes en la base de datos completa en una sola pasada
- Es posible que no se encuentran todos los conjuntos frecuentes (falsos negativos)
  - Se pueden reducir si disminuimos el soporte mínimo en la muestra; por ej.  $0.9 \cdot \text{minsup}$ .

1. Se divide la base de datos en trozos y se aplica el algoritmo apriori, PCY, multietapa o multihash a cada trozo
  - Se toman como candidatos los conjuntos frecuentes encontrados en uno o más trozos
  - Si un conjunto no es frecuente en ningún trozo, su soporte es menor a *minsup* en el conjunto de todos los trozos y en general en la base de datos completa
2. En una segunda pasada se descartan los candidatos que no sean frecuentes en el conjunto completo de transacciones

## ALGORITMO DE SON EN EL MODELO MAPEO-REDUCCIÓN

1. Mapeo 1: Toma el subconjunto de transacciones asignadas y encuentra los conjuntos frecuentes con Apriori, PCY, multietapa o multihash
2. Reducción 1: Produce los conjuntos frecuentes que aparecen al menos una vez como candidatos
3. Mapeo 2: Recibe todos los candidatos y cuenta sus ocurrencias en una porción de la base de datos
4. Reducción 2: Toma cada candidato y suma sus ocurrencias en distintas porciones para producir su soporte en la base de datos completa

## ALGORITMO DE TOIVONEN

- Se selecciona una muestra de la base de datos para encontrar candidatos usando un soporte mínimo más pequeño (por ej.  $0.9 \cdot p \cdot \text{minsup}$ )
- Agrega la frontera negativa al conjunto de candidatos
  - La frontera negativa son todos los conjuntos que no son frecuentes en la muestra pero cuyos subconjuntos inmediatos<sup>1</sup> sí lo son
- Se cuentan los candidatos en la base de datos completa
  1. Ningún miembro de la frontera negativa es frecuente: los conjuntos frecuentes son los encontrados en la muestra
  2. Algunos miembros de la frontera negativa son frecuentes:  
no es posible dar una respuesta

---

<sup>1</sup>Subconjuntos con un elemento menos