ALGUNAS ESTRUCTURAS DE DATOS PROBABILÍSTICOS : BLOOM FILTER, COUNT-MIN SKETCH E HYPERLOGLOG

1. RESUMEN

Este artículo describe tres estructura de datos probabilísticos : Bloom Fitler , Count-min sketch e HyperLogLog , programaremos cada uno en lenguaje R y comprobaremos su funcionamiento para el conteo y las consultas referidas a elementos dentro de un conjunto.

2. INTRODUCCIÓN

2.1. Objetivos

- Usar la estructura bloom Fitler para hacer consultas sobre elementos que puedas pertenecer a la estructura.
- Usar la estructura Count-min sketch.
- Usar la estructura HyperLogLog para calcular el número aproximado de elementos en un multiconjunto.

En esta oportunidad usaremos cada una de las tres estructuras . En el primer (Bloom Filter) generaremos un conjuntos de n elementos aleatorios , seguidamente utilizaremos las funciones añadir y query(consultar) para añadir elementos y consultar si algún elemento pertenece al conjunto que hemos generado en el caso de la estructura Count-min sketch , finalmente con la estructura HyperLogLog generaremos un multiconjunto grande el cual calcularemos al cardinalidad de un elemento específico con una excatitud del 2 % usando 1,5kb de memoria.

3. BLOOM FITLER

- Estructura de datos espacio eficiente (Borton Howard Bloom - 1970).
- Usado para probar si un elemento es miembro de un conjunto.
- Una consulta arroja “posiblemente en el conjunto”(falso positivo) o “definitivamente no en el conjunto”(falso negativo).
- Se puede agregar elementos al conjunto pero no removerlos.
- Mayores elementos con añadidos , mayores probabilidad de falsos positivos .

- Generalmente , menos de 10 bits por elemento son requeridos para un 1 % de probabilidad falso positivo , independientemente del tamaño o número de elementos del set .

3.1. Descripción del algoritmo

Arreglo de m bits , todo en 0 . También debe haber k funciones hash , cada uno de los cuales mapea algún elemento a una de las m posiciones (distribución aleatoria uniforme).

3.2. Para añadir un elemento

Se alimenta a cada una de las k funciones para obtener k posiciones en el arreglo . Todas esas posiciones se van a 1 .

3.3. Para consultar (si es que el elemento está)

Se alimenta a cada una de las k funciones para obtener k posiciones en el arreglo , si algunas de las posiciones son 0 el elemento definitivamente no se encuentra en el conjunto , si todos son 1, entonces o bien el elemento está o los bits han sidos puestos a 1 cuando fueron insertados.

4. COUNT-MIN SKETCH

- Sirve como una tabla de frecuencia de eventos en una data stream . Usa funciones hash para mapear eventos a frecuencias , pero solo usa espacios sub-lineales a costa de "sobrecontar algunos eventos a causa de colisiones (2003-Graham Cormode , S.Muthu Muthokrishnan).
- Esencialmente es lo mismo que Bloom fitler, pero son usados de manera diferente y se ponen el tamaño de manera diferente.

4.1. Estructura

El objetivo de count-min sketch es la de consumir un stream de eventos , uno a la vez , y contar la frecuencia de los diferentes tipos de eventos en el stream . En cualquier momento el sketch puede ser consultado para la frecuencia de un particular tipo de evento y regresará un estimado de esta frecuencia dentro de una cierta distancia de la frecuencia real , con una cierta probabilidad.

5. HYPERLOGLOG

- Algoritmo para el conteo distintivo , aproximando el número de distintos elementos en un multiconjunto . Es capaz de estimar cardinalidades mayor a 10^9 con una certeza del 2 % usando 1,5 kb de memoria.

5.1. Algoritmo

Se basa en la observación de que las cardinalidades de un multiconjunto de números aleatorios uniformemente distribuidos pueden ser calculados estimando el máximo número de ceros en la representación binaria de cada número en el conjunto . Si el máximo número de ceros observados es n , un estimado para el número de elementos en el conjunto es 2^n .

5.2. Operaciones

1. Add: Añade un elenento al conjunto.
2. Count: Nos arroja la cardinalidad el conjunto.
3. Merge: Para obtener la unión de dos conjuntos