Data Streams:

Estimating Moments

Mining Massive Datasets

Prof. Carlos Castillo — https://chato.cl/teach

Universitat Pompeu Fabra Barcelona

Sources

- Mining of Massive Datasets (2014) by Leskovec et al. (chapter 4)
  - Slides part 1, part 2
- Tutorial: Mining Massive Data Streams (2019) by Michael Hahsler

Estimating moments

Moments of order k

- If a stream has A distinct elements, and each element has frequency m_i
- The kth order moment of the stream is $$\sum_i m_i^k$$
- **The 0th order moment** is the number of distinct elements in the stream
- **The 1st order moment** is the length of the stream

Moments of order k (cont.)

- The k^{th} order moment of the stream is $$\sum_i m_i^k$$
- **The 2nd order moment** is also known as the "surprise number" of a stream (large values = more uneven distribution)

$\sum m_i^2$

m_i	i=1	i=2	i=3	i=4	i=5	i=6	i=7	i=8	i=9	i=10	i=11	2nd moment
Seq1	10	9	9	9	9	9	9	9	9	9	9	910
Seq2	90	1	1	1	1	1	1	1	1	1	1	8110

Method for second moment

- Assume (for now) that we know n, the length of the stream
- We will sample s positions
- For each sample we will have $X.element$ and $X.count$
- We sample s random positions in the stream

  $X.element$ = element in that position, $X.count \leftarrow 1$

  When we see $X.element$ again, $X.count \leftarrow X.count + 1$

- Estimate second moment as $n(2 \times X.count - 1)$

Alon, N., Matias, Y., & Szegedy, M. (1999). The space complexity of approximating the frequency moments. Journal of Computer and system sciences, 58(1), 137-147.

Method for second moment (cont.)

- Example: a,b,c,b,d,a,c,d,a,b,d,c,a,a,b

  $m_a = 5,\ m_b = 4,\ m_c = 3,\ m_d = 3$

  second moment = $5^2+4^2+3^2+3^2 = \mathbf{59}$

- Suppose we sample s=3 variables X_1, X_2, X_3
- Suppose we pick the 3rd, 8th, and 13th position at random
- X_1.element=c, X_2.element=d, X_3.element=a
- X_1.count=3, X_2.count=2, X_3.count=2 (we count forwards only!)
- **Estimate** $n(2 \times X.count - 1)$, first estimate = 15(6-1) = 75,

  second estimate 15(4-1) = 45, third estimate 15(4-1) = 45,

  average of estimates = 55≃**59**

Alon, N., Matias, Y., & Szegedy, M. (1999). The space complexity of approximating the frequency moments. Journal of Computer and system sciences, 58(1), 137-147.

Method for second moment (cont.)

- Example: a, b, c, b, d, a, c, d, a, b, d, c, a, a, b
- Suppose we pick the 3rd, 8th, and 13th position at random
- X_1.element=c, X_2.element=d, X_3.element=a
- X_1.count=3, X_2.count=2, X_3.count=2

Alon, N., Matias, Y., & Szegedy, M. (1999). The space complexity of approximating the frequency moments. Journal of Computer and system sciences, 58(1), 137-147.

Why this method works?

- Let e(i) be the element in position i of the stream
- Let c(i) be the number of times e(i) appears in positions i, i+1, i+2, ..., n
- Example: `a,b,c,b,d,a,c,d,a,b,d,c,a,a,b`

  `c(6) = ?`

Why this method works?

- Let e(i) be the element in position i of the stream
- Let c(i) be the number of times e(i) appears in positions i, i+1, i+2, ..., n
- Example: a,b,c,b,d,<u>a</u>,c,d,<u>a</u>,b,d,c,<u>a</u>,<u>a</u>,b

  c(6) = 4 (remember: we count forwards only!)

Why this method works? (cont.)

- c(i) is the number of times e(i) appears in positions i, i+1, i+2, …, n
- E[$n\,(2 \times X.count - 1)$] is the average of $n\,(2\,c(i) - 1)$ over all positions $i=1..n$

$$E[n(2 \times X.\mathrm{count} - 1)] = \frac{1}{n}\sum_{i=1}^{n} n(2c(i) - 1)$$

$$E[n(2 \times X.\mathrm{count} - 1)] = \sum_{i=1}^{n} (2c(i) - 1)$$

Why this method works? (cont.)

$$E[n(2 \times X.\text{count} - 1)] = \sum_{i=1}^{n} (2c(i) - 1)$$

- Now focus on element a that appears m_a times in the stream
  - The last time a appears this term is 2c(i) – 1 = 2x1-1 = 1
  - Just before that, 2c(i)-1 = 2x2-1 = 3
  - …
  - Until $2m_a$ – 1 for the first time a appears
- Hence

$$E[n(2 \times X.\text{count} - 1)] = \sum_{a} 1 + 3 + 5 + \cdots + (2m_a - 1) = \sum_{a} m_a^2$$

For higher order moments (v = X.count)

- For second order moment
  - We use $n(2v\text{-}1) = n(v^2 - (v\text{-}1)^2)$
- For third order moment
  - We use $n(3v^2 - 3v + 1) = n(v^3 - (v\text{-}1)^3)$
- For k^{th} order moment
  - We use $n(v^k - (v\text{-}1)^k)$

For infinite streams

- Use a reservoir sampling strategy
- If we want s samples
  - Pick the first s elements of the stream setting X_i.element $\leftarrow$ e(i) and X_i.count $\leftarrow 1$ for i=1...s
  - When element n+1 arrives
    - Pick X_{n+1}.element with probability s/(n+1), evicting one of the existing elements at random and setting X.count $\leftarrow 1$
- As before, probability of an element is s/n

Summary

Things to remember

- k^{th} order moments of a stream

Exercises for TT22-T26

- Mining of Massive Datasets (2014) by Leskovec et al.
  - Exercises 4.2.5
  - Exercises 4.3.4
  - Exercises 4.4.5
  - Exercises 4.5.6