Data Streams:

Reservoir Sampling

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

Sampling a fixed-size sample

A fixed-size sample

- We normally do not know the stream size
- We just know how much storage space we have
- Suppose we have storage space s and want to maintain a random sample S of size $s = |S|$
- **Requirement**: after seeing n items, each of the n items should be in our sample with probability s/n
  - *No item should have an advantage or disadvantage*

Bad solutions

- Suppose stream $= <$ a, f, e, b, g, r, u, ... $>$
- **Requirement**: after seeing n items, each of the n items should be in our sample with probability s/n
- Suppose s=2
  - Always keep first 2? No, because then p(a) $= 1 \neq 0 =$ p(e)
  - Always keep last 2? No, because then p(a) $= 0 \neq 1 =$ p(u)
- Sample some ... which? Then evict some ... which?

Reservoir sampling

(one of the most beautiful algorithms of this course)

- Elements $x_1, x_2, x_3, \ldots, x_i, \ldots$
- Store all first s elements $x_1, x_2, \ldots, x_s$
- Suppose element x_n arrives
  - With probability $1 - s/n$, ignore this element
  - With probability s/n:
    - Discard a random element from the reservoir
    - Insert element x_n into the reservoir

Exercise

RESERVOIR SAMPLING

Store all first s elements x_1, x_2, …, x_s

When element x_n arrives

- With probability $1 - s/n$, ignore
- With probability s/n:
  - Discard randomly from reservoir
  - Insert element x_n into the reservoir

- Suppose input is <a, b, c, …>
- Suppose s = 2
- Suppose we just processed element "c"
- What is:
  - Probability "a" is in the sample?
  - Probability "b" is in the sample?
  - Probability "c" is in the sample?
- If you are done quickly,

  try one more element, "d"

Answer in
Nearpod Open-Ended Question

Proof by induction

- **Inductive hypothesis**: after n elements seen

  each of them is sampled with probability s/n

- **Inductive step**: element x_{n+1} arrives,
  - what is the probability than an already-sampled element x_i stays in the sample?

$$\underbrace{\left(1 - \frac{s}{n+1}\right)}_{x_{n+1}\text{ not sampled}} + \underbrace{\left(\frac{s}{n+1}\right)}_{x_{n+1}\text{ sampled}} \cdot \underbrace{\left(\frac{s-1}{s}\right)}_{x_i\text{ not evicted}} = \frac{n}{n+1}$$

Proof by induction (cont.)

- Tuple x_{n+1} is sampled with probability $\dfrac{s}{n+1}$ ✓
- Tuples x_i with $i \le n$
  - Were in the sample with probability s/n
  - Stay in the sample with probability $n/(n+1)$
  - Hence, are in the sample with probability

$$\frac{s}{n} \cdot \frac{n}{n+1} = \frac{s}{n+1}$$ ✓

Recency-biased reservoir sampling

- Before we had $p(i) = s/n$
  - Probability of element x_i to be included
  - Reservoir of size s
  - Stream so far of size n
- Suppose we want a different $p(i) \propto f(i,n)$
  - Example: $f(i,n)$ larger for more recent items

Data Mining, The Textbook (2015) by Charu Aggarwal (section 12.2.1.1. “Handling Concept Drift”)

Recency-biased reservoir sampling (cont.)

- Suppose we want $p(i) \propto f(i,n) = e^{-\lambda(n-i)}$
- Parameter $\lambda \in [0,1]$ is a decay factor and $s < \frac{1}{\lambda}$
- Algorithm: reservoir starts empty

  At time n, it is $F(n) \in [0,1]$ full

  x_{n+1} arrives and is inserted with probability $\lambda \cdot s$

  If x_{n+1} is inserted, remove from reservoir a random element with probability $F(n)$

See proof in: Aggarwal, C. (2006). On biased reservoir sampling in the presence of stream evolution. Proc. VLDB

The longer an item is in the reservoir, the more likely is this item to be evicted.

Summary

Things to remember

- How to do reservoir sampling
- How to compute probabilities in reservoir sampling, i.e., how to prove it's correct

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