Big Data Analytics

Vidyalankar Institute of Technology
Accredited A+ by NAAC

Academic Year 2022-23

Agenda

1. Streaming Algorithms
2. Bloom Filter
3. Flajolet-Martin Algorithm
4. Datar-Gionis-Indyk-Motwani Algorithm

Prof. Prakash Parmar
Data Engineer | Data Analyst
Assistant Professor
Vidyalankar Institute of Technology, Mumbai
GATE Instructor: Vidyalankar Classes

Streaming Algorithms

Vidyalankar Institute of Technology
Accredited A+ by NAAC

- Sampling
- Filtering Data- Blooms filter
- Count Distinct Elements: Flajolet-Martin Algorithm
- Sliding Window Algorithms

Streaming Algorithms

VIT Vidyalankar Institute of Technology
Accredited A+ by NAAC

Sampling Algorithms:

Description: Sampling algorithms select a representative subset of data points from a data stream for analysis. They provide an approximate view of the data while reducing computational requirements.

Example: Random sampling or reservoir sampling algorithms are used to select a random sample of data points for estimating statistical properties, such as mean and variance.

Filtering Algorithms:

Description: Bloom filters are probabilistic data structures that efficiently test set membership. They are used for identifying whether an element has been seen in the stream.

Example: Bloom filters are employed in stream processing to reduce the number of unnecessary lookups in databases or storage systems.

Counting Algorithms:

Description: Counting algorithms estimate the cardinality (number of distinct elements) of a data stream efficiently. They are useful for counting unique items in large data streams.

Example: The Flajolet-Martin algorithm and HyperLogLog are commonly used counting algorithms for estimating distinct elements in a stream.

Streaming Algorithms

VIT Vidyalankar Institute of Technology
Accredited A+ by NAAC

Sliding Window Algorithms:

Description: Sliding window algorithms maintain a fixed-size window over the data stream, allowing for the analysis of recent data while discarding older data.

Example: Sliding window aggregates, like sliding window averages or sliding window counts, are used for real-time monitoring and trend analysis.

Streaming Machine Learning Algorithms:

Description: Streaming machine learning algorithms adapt traditional machine learning models to work with streaming data. They continuously update models as new data arrives.

Example: Online learning algorithms like stochastic gradient descent (SGD) or streaming versions of popular machine learning models (e.g., streaming k-means) are used for streaming analytics.

Change Detection Algorithms:

Description: Change detection algorithms monitor data streams for significant changes or anomalies. They trigger alerts or actions when unusual patterns are detected.

Example: Cumulative sum (CUSUM) and Exponentially Weighted Moving Average (EWMA) are used for change detection in streaming data.

Bloom Filter

VIT Vidyalankar Institute of Technology
Accredited A+ by NAAC

- A Bloom filter is a probabilistic data structure used for testing the membership of an element in a set.
- It uses multiple hash functions and a bit array to represent the set and perform membership tests.
- It's particularly useful when you want to quickly check if an item is part of a larger dataset without storing the entire dataset

Common use cases for Bloom filters

VIT Vidyalankar Institute of Technology
Accredited A+ by NAAC

Caching Systems:

To quickly check whether a web page, query result, or data record is already in a cache before performing a more resource-intensive retrieval.

Use Case: Web caching, database caching, and content delivery networks (CDNs) employ Bloom filters to quickly check whether a requested web page, object, or data record is already in the cache.

Benefits: Reduces the need to fetch data from slower storage, improving response times and reducing network and server load.

Spam and Malware Filtering:

To identify whether an incoming email address or message sender is in a list of known spammers.

Use Case: Email servers and security systems use Bloom filters to determine if incoming email addresses, sender domains, or message content matches known spam or malware patterns.

Benefits: Efficiently identifies and blocks spam emails and malicious content, reducing the risk of malware infections and protecting users from unwanted messages.

Common use cases for Bloom filters

Vidyalankar Institute of Technology
Accredited A+ by NAAC

Network Packet Filtering:

To identify whether an IP address is in a list of known malicious or blocked addresses.

Use Case: Firewalls and intrusion detection systems (IDS) employ Bloom filters to quickly check whether incoming or outgoing network packets match known blacklists or threat signatures.

Benefits: Helps in identifying and blocking network traffic from malicious IP addresses or domains, enhancing network security.

URL Shorteners:

Use Case: URL shortening services use Bloom filters to check whether a given shortened URL has already been created.

Benefits: Ensures that unique shortened URLs are generated, avoiding duplication in the mapping between long URLs and short codes.

Recommendation Systems:

Use Case: Online recommendation systems utilize Bloom filters to determine whether a user has already interacted with or rated a specific item or content.

Benefits: Enhances user experience by preventing repetitive recommendations and promoting new or relevant content.

Bloom Algorithm

VIT Vidyalankar Institute of Technology
Accredited A+ by NAAC

Initialization:

Create a bit array of length 'm' and initialize all bits to 0.

Choose 'k' different hash functions.

Adding an Element:

To add an element 'x' to the Bloom filter, apply each of the 'k' hash functions to 'x' to obtain 'k' hash values.

Use each of these 'k' hash values to index into the bit array, setting the corresponding bits to 1.

Checking Membership:

To check whether an element 'y' is a member of the set represented by the Bloom filter, apply each of the 'k' hash functions to 'y' to obtain 'k' hash values.

Check the bit at each of the 'k' indices in the bit array. If all of them are set to 1, the Bloom filter suggests that 'y' may be in the set. However, there's a possibility of false positives due to hash collisions.

Bloom Algorithm: Example-1

Vidyalankar Institute of Technology
Accredited A+ by NAAC

Use a small-sized Bloom filter with a bit array of length 10 and two hash functions.

Step 1: Initialize the Bloom Filter

Bit array length (m): 10 (meaning we have 10 bits in our filter)

Number of hash functions (k): 2 **Bit Array: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]**

Step 2: Adding Elements Let's add three numerical elements to the Bloom filter: 7, 14, and 21.

Element 7:

Hash Function 1: h1(7) = 7 % 10 = 7

Hash Function 2: h2(7) = (7 * 3) % 10 = 21 % 10 = 1

Set the bits at positions 7 and 1 to 1. **Bit Array: [0, 1, 0, 0, 0, 0, 0, 1, 0, 0]**

Element 14:

Hash Function 1: h1(14) = 14 % 10 = 4

Hash Function 2: h2(14) = (14 * 3) % 10 = 42 % 10 = 2

Set the bits at positions 4 and 2 to 1. **Bit Array: [0, 1, 1, 0, 1, 0, 0, 1, 0, 0]**

Bloom Algorithm: Example-1

Vidyalankar Institute of Technology
Accredited A+ by NAAC

Element 21:

Hash Function 1: h1(21) = 21 % 10 = 1

Hash Function 2: h2(21) = (21 * 3) % 10 = 63 % 10 = 3

Set the bits at positions 1 and 3 to 1. **Bit Array: [0, 1, 1, 1, 1, 0, 0, 1, 0, 0]**

Step 3: Checking Membership, let's check for membership of two numerical values: 7 and 12.

Checking for 7:

Hash Function 1: h1(7) = 7 % 10 = 7

Hash Function 2: h2(7) = (7 * 3) % 10 = 21 % 10 = 1

Both bits at positions 7 and 1 are 1, indicating that 7 might be in the set.

Checking for 12:

Hash Function 1: h1(12) = 12 % 10 = 2

Hash Function 2: h2(12) = (12 * 3) % 10 = 36 % 10 = 6

Neither bit at positions 2 nor 6 is 1, indicating that 12 is not in the set.

Bloom Algorithm: Example-1

Vidyalankar Institute of Technology
Accredited A+ by NAAC

Results:

7 is indicated as possibly being in the set (a false positive).

12 is correctly indicated as not being in the set.

It's important to note that Bloom filters can produce false positives, as shown in this example.

When the bits for multiple elements overlap due to hash collisions, it can lead to false positives. However, they never produce false negatives, meaning that if the filter indicates an element is not in the set, it is indeed not in the set.

Bloom Algorithm: Example-2

Vidyalankar Institute of Technology
Accredited A+ by NAAC

Consider size of Bloom filter m = 5

Two hash functions: h1(x) = x mod 5 and h2(x) = (2x+6) mod 5

Insert element 10 and 7 in the bloom filter of size 5.

Element to insert	h1(x) = x mod 5	h2(x) = (2x+6) mod 5	Bloom filter
			0 0 0 0 0
10	10 % 5 = 0	(2*10+6) % 5 = 1	1 1 0 0 0
7	7%5 = 2	(2*7+6) % 5 = 0	1 1 1 0 0

Bloom Algorithm: Example-2

Vidyalankar Institute of Technology
Accredited A+ by NAAC

Bloom filter status

1	1	1	0	0

Check, Are elements 14 and 15 presents?

Element to check	h1(x) = x % 5	h2(x) = (2x+6) % 5	Presence?
14	14 % 5 = 4	(2*14+6) % 5 = 4	At index 4, the value is 0 which means that the element is not present in the set, accurately says absence.
15	15 % 5 = 0	(2*15+6) % 5 = 1	At both index 0 and 1, the value is 1 which means depicts that element is present in the set, but in reality, it is not (False Positive). (Only provides probability of presence)

Learn Fundamentals & Enjoy Engineering

Thank You

Prof. Prakash Parmar
Data Engineer | Data Analyst
Assistant Professor
Vidyalankar Institute of Technology, Mumbai
GATE Instructor: Vidyalankar Classes