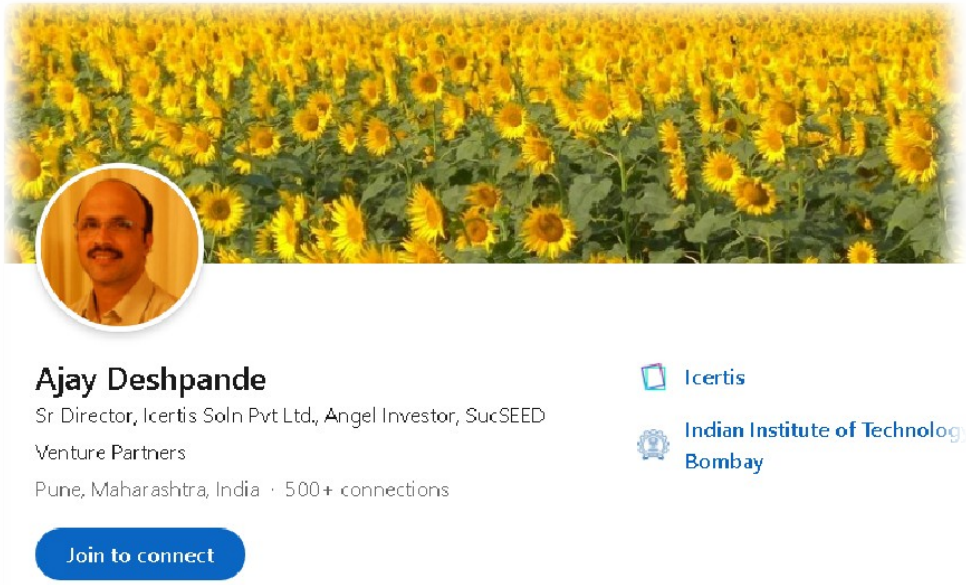
Data Streaming

Ajay Deshpande
Linkedin.com/in/deshpandeajay
dajay0@yahoo.com



The Syllabus: Data Science

Data Streams: Stream data model, stream sources, stream queries, issues in stream processing, sampling data in a stream, stream filtering: bloom filter

[6 Hrs]

Text Books

- "Mining of Massive Datasets", Jure Leskovec, Anand Rajaraman, and Jeffery David Ullman, Cambridge University Press, 2 edition (13 November 2014) , ISBN-10: 1107077230, ISBN-13: 978-1107077232
- "Data Mining: Concepts and Techniques", Jiawei Han, Micheline Kamber, 3rd Edition, Morgan Kaufmann, ISBN-13: 978-9380931913

What is Data Mining?

Data Mining of Streams'

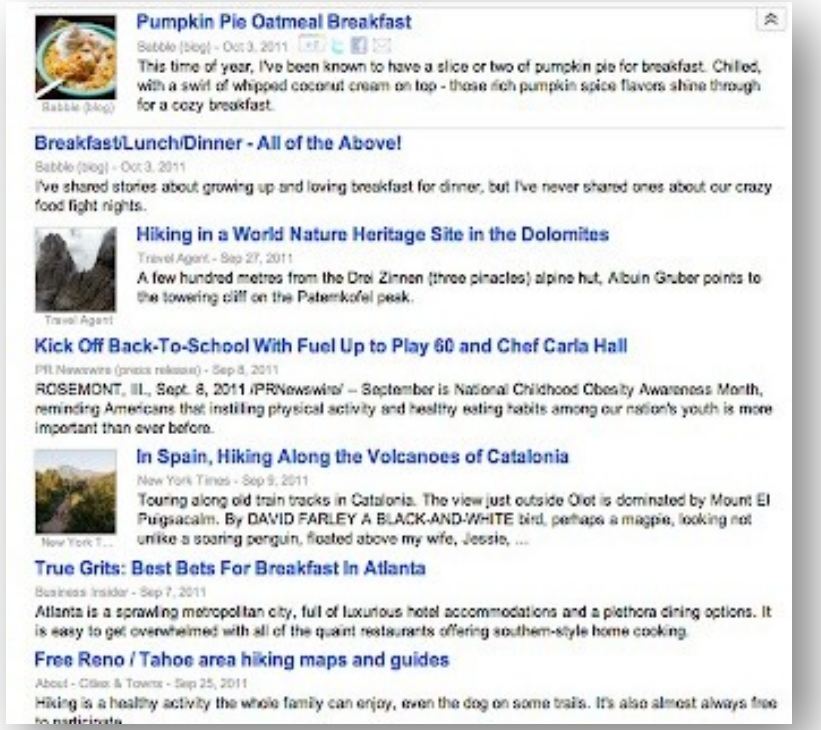
Let’s Solve: Top Read Articles on TOI

News Readers

Times of India Website

Click: toi.com/x/y/x

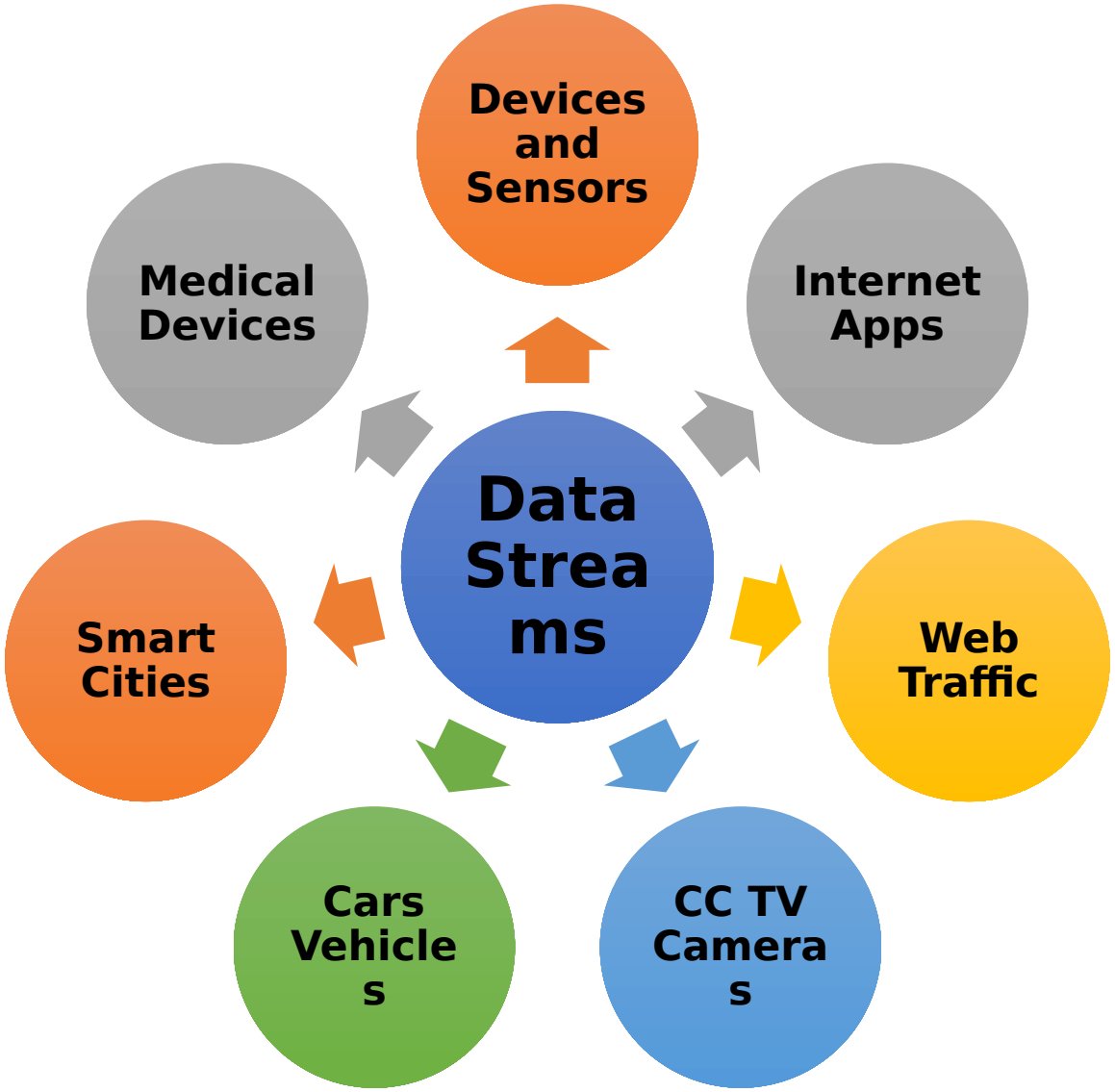
Portal of Top Articles



So What are Streams

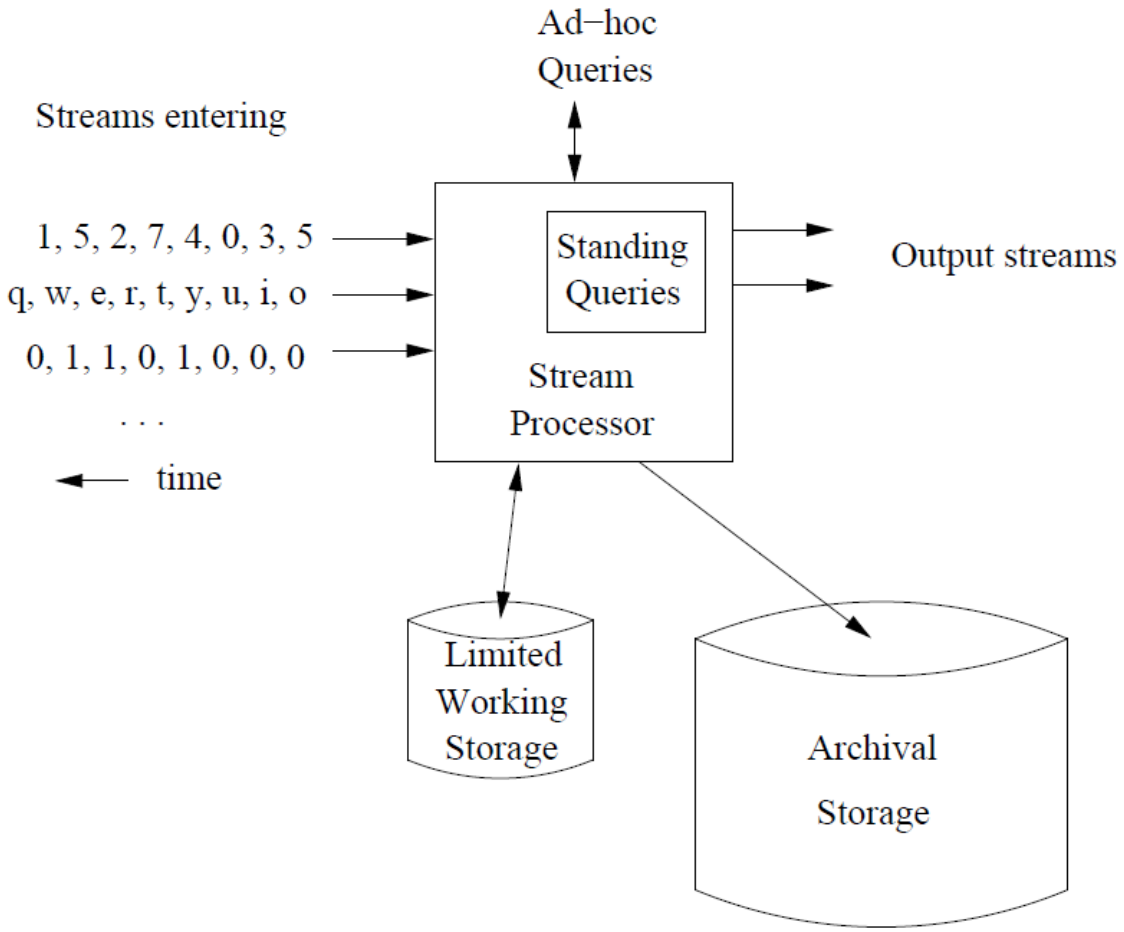
- Rapid Rate of Data flow on inputs
- The Data Processing System does cannot control the inputs
- Typically these are infinite streams and can come from multiple sources
- Data cannot be processed **after** all the data has arrived
- Comparing with Media Streams
  - Although these streams are large, they are finite (End of Movie!)
  - These streams are **played** but not inferred from: no conclusions derived
- What applications are we talking about?

Examples of Streaming Data Sources



And Many More!

The Streams Data Model



A Data Stream Management System

Salient Features

- Multiple Input (usually non-uniform) Streams
- Archival Storage (like Tapes)
- Working Storage
- Standing and Ad-hoc Queries

Issues in Stream Processing

- High Rate and volume
  - Cannot be stored for long
  - Data Lost if not processed immediately
  - One system typically caters for many streams at once
- Approach will need to aggregate as data arrives
  - Cannot **wait for all the data to come**
  - Practical Approximation vs Exact
- Can consider a **Window** of the last n elements
  - Sliding vs Jumping Windows
  - Window size of 1 is the trivial case
- Need ways to identify and discard unwanted data: Filtering
- Hashing based Algorithms well suited for Stream Processing

Sampling of Data in Streams

- Lets assume we have a stream of Google Searches
  U3, Search String, Time
  U8, Search String, Time

- Various possible ways to sample this stream
- Remember Sampling leads to approximations in answers
- What are the aspects you will consider to sample this stream?

How would you answer

- What fraction of a **typical user's** Searches were repeated in the last month?
- How many Unique Users are on the site?

- The first query considers only Users whereas the second one spans users and their Search Strings
- Be aware: estimates computed depend on the sampling technique used

How can Sampling Help

- Question: What fraction of a **typical user's** Searches were repeated in the last month?
- Note: we cannot answer this by counting duplicates in the searches
- Brute force solution:
  - Create a list of users and their searches
  - For every new tuple in the stream update this list if a search is repeated
  - Needs large amounts of storage
- Sampling approach
  - Let us target to capture the Searches of 1/10th of the users
  - Map user name to [1-10] using Hashing. Consider only users who hash to 1
  - Store these users and Search strings for them
  - Hashing ensures that I capture samples of the same set of users without much computation
  - Using this data I can estimate the answer…

Using Bitmaps to our Advantage

- Question: How many Unique Users are on the site?
- Brute force solution:
  - Keep a list of users
  - For every new tuple in the stream add to this list if user is first timer
  - Update unique count
- Sampling approach
  - Assume that there are N total users (N can be a large number)
  - Have a BitMap of N bits
  - Hash the incoming user name to a number, X which is [1-N]
  - Set BitMap[X] = 1
  - Repeat for a Sampling period (say 1 week)
  - From this data we have Count of logins and % of unique users
  - Can be extrapolated to the year

Example: Lets Build a Spam Filter

- Problem Definition:
  - You need to build a filter for the email gateway – Allow or Reject emails
  - Have 1 Billion ‘allowed’ addresses
  - Emails from all other addresses are to be rejected
- Solution
  - Lets use 1 GB of ram as a bit array => 8 Billion Bits
  - Use a Hashing function Email Addrs => 8 Billion Addresses
    - Set the bits for allowed email addresses to 1
  - When stream email arrives, hash the sender’s email address as above
  - If Bit is set to 1 allow the email in
- Do you think we will let in some spam emails?
  - False positives
- We just implemented the simple Bloom Filter
  - First conceived by Burton Howard Bloom

Formalizing the Bloom Filter

- Definition
  - S is a set of keys (lets say m)
  - An array of n bits – n usually larger than m
    - Initially all bits are 0
  - A set of hash functions h(K)
    - We set A[h(K)] is set to 1 for all K that need to be let in
- We can use multiple hash functions to increase accuracy
  - Hashing into the same array or different arrays (more storage)
- Bloom filtering is not dependent on the size of the input (constant time)

Recap of what we studied

- Data Streams
  - Distinction between Web Streams and Streams in Data Science
- Data Mining in Streams
- The Stream data model
  - Sources of Streams
  - Queries: Standing and Ad Hoc
- Issues to consider in stream processing
- How can Sampling help in a processing a stream
- Filtering data in Streams
  - The Bloom filter and why it can generate false positives

THANK YOU !!

Ajay Deshpande
Linkedin.com/in/deshpandeajay
dajay0@yahoo.com