B609 Sublinear Algorithms for Big Data

Qin Zhang

Now about the Big Data

- Big data is everywhere
  - Walmart: over 2.5 petabytes of sales transactions
  - Google : an index of over 19 billion web pages
  - facebook: over 40 billion of pictures
  - . . .

Now about the Big Data

- Big data is everywhere
  - Walmart: over 2.5 petabytes of sales transactions
  - Google : an index of over 19 billion web pages
  - facebook: over 40 billion of pictures
  - . . .
- Magazine covers

Nature '06

Nature '08

CACM '08

Economist '10

Source and Challenge

- **Source**
  - Retailer databases
  - Logistics, financial & health data
  - Social network
  - Pictures by mobile devices
  - Internet of Things
  - New forms of scientific data

Source and Challenge

- Source
  - Retailer databases
  - Logistics, financial & health data
  - Social network
  - Pictures by mobile devices
  - Internet of Things
  - New forms of scientific data
- Challenge
  - Volume
  - Velocity
  - Variety (Documents, Stock records, Personal profiles, Photographs, Audio & Video, 3D models, Location data, . . .)

Source and Challenge

- Source
  - Retailer databases
  - Logistics, financial & health data
  - Social network
  - Pictures by mobile devices
  - Internet of Things
  - New forms of scientific data

- Challenge
  - Volume
  - Velocity

  } The focus of algorithm design

  - Variety (Documents, Stock records, Personal profiles, Photographs, Audio & Video, 3D models, Location data, . . .)

What is the meaning of Big Data IN THEORY?

- We don't define Big Data in terms of TB, PB, EB, . . .

What is the meaning of Big Data IN THEORY?

- We don't define Big Data in terms of TB, PB, EB, . . .
- The data is stored there, but no time to read them all.
  What can we do?

What is the meaning of Big Data IN THEORY?

- We don't define Big Data in terms of TB, PB, EB, . . .
- The data is stored there, but no time to read them all.
  What can we do?
  - Read some of them.

Sublinear in time

What is the meaning of Big Data IN THEORY?

- We don't define Big Data in terms of TB, PB, EB, ...
- The data is stored there, but no time to read them all.
  What can we do?
  - Read some of them.

Sublinear in time

- The data is too big to fit in main memory.
  What can we do?

What is the meaning of Big Data IN THEORY?

- We don't define Big Data in terms of TB, PB, EB, ...
- The data is stored there, but no time to read them all.
  What can we do?
  - Read some of them. Sublinear in time
- The data is too big to fit in main memory.
  What can we do?
  - Store on the disk (page/block access) Sublinear in I/O

What is the meaning of Big Data IN THEORY?

- We don't define Big Data in terms of TB, PB, EB, . . .
- The data is stored there, but no time to read them all.
  What can we do?
  - Read some of them. Sublinear in time
- The data is too big to fit in main memory.
  What can we do?
  - Store on the disk (page/block access) Sublinear in I/O
  - Throw some of them away. Sublinear in space

What is the meaning of Big Data IN THEORY?

- We don't define Big Data in terms of TB, PB, EB, . . .
- The data is stored there, but no time to read them all.
  What can we do?
  - Read some of them. Sublinear in time
- The data is too big to fit in main memory.
  What can we do?
  - Store on the disk (page/block access) Sublinear in I/O
  - Throw some of them away. Sublinear in space
- The data is too big to be stored in a single machine.
  What can we do if we do not want to throw them away?

What is the meaning of Big Data IN THEORY?

- We don't define Big Data in terms of TB, PB, EB, ...
- The data is stored there, but no time to read them all.
  What can we do?
  - Read some of them. Sublinear in time
- The data is too big to fit in main memory.
  What can we do?
  - Store on the disk (page/block access) Sublinear in I/O
  - Throw some of them away. Sublinear in space
- The data is too big to be stored in a single machine.
  What can we do if we do not want to throw them away?
  - Store in multiple machines, which collaborate via communication

    Sublinear in communication

What do we mean by "sublinear"?

Time/space/communication spent is o(input size)

Conceretly, theory folks talk about the following ...

- Sublinear time algorithms
  - Sublinear time approximation algorithms
  - Property testing (not in this course)

Concerетly, theory folks talk about the following ...

- Sublinear time algorithms
  - Sublinear time approximation algorithms
  - Property testing (not in this course)
- Sublinear space algorithms
  - Data stream algorithms

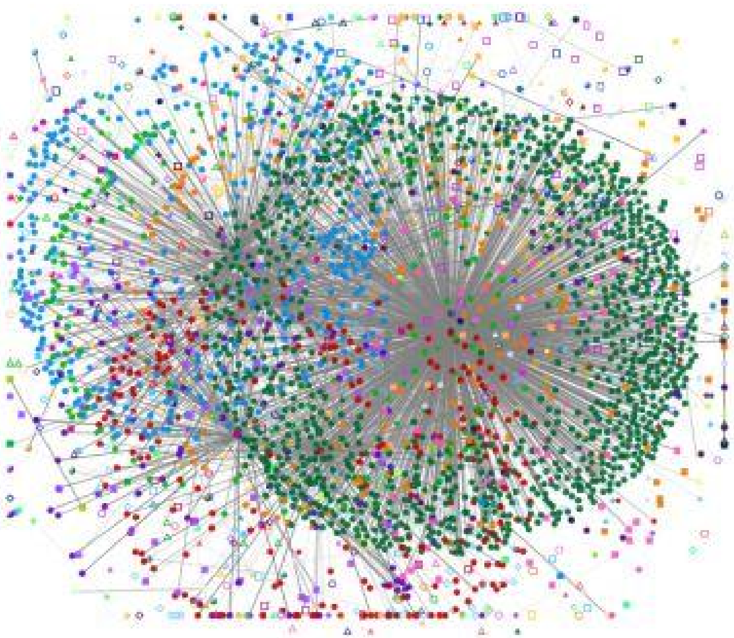
Conceretly, theory folks talk about the following ...

- Sublinear time algorithms
  - Sublinear time approximation algorithms
  - Property testing (not in this course)
- Sublinear space algorithms
  - Data stream algorithms
- Sublinear communication algorithms
  - Multiparty communication protocols/algorithms (particular models: MapReduce, BSP, . . .)

ConcerETLY, theory folks talk about the following ...

- Sublinear time algorithms
  - Sublinear time approximation algorithms
  - Property testing (not in this course)
- Sublinear space algorithms
  - Data stream algorithms
- Sublinear communication algorithms
  - Multiparty communication protocols/algorithms (particular models: MapReduce, BSP, ...)
- Sublinear I/O algorithms (not in this course)
  - External memory data structures/algorithms

Sublinear in time



Example:

Given a social network graph, we want to compute its **average degree**.
(i.e., the average # of friends people have in the network)

Can we do it without quering the degrees of all nodes?
(i.e., asking everyone)

Why hard? You can't see everything in sublinear time!

- Computing exact average degree is impossible without querying at least $n-1$ nodes (n: $\#$ total nodes).

  So our goal is to get a $(1+\epsilon)$-approximation w.h.p.
  (ϵ is a very small constant, e.g., 0.01)

Why hard? You can't see everything in sublinear time!

- Computing exact average degree is impossible without querying at least $n - 1$ nodes (n: # total nodes).

  So our goal is to get a $(1 + \epsilon)$-approximation w.h.p.
  (ϵ is a very small constant, e.g., 0.01)

- Can we simply use sampling?

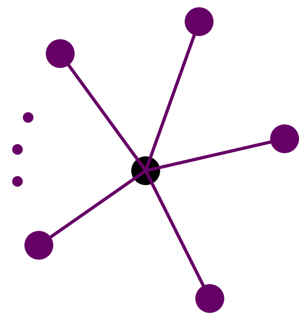
Why hard? You can't see everything in sublinear time!

- Computing exact average degree is impossible without querying at least $n-1$ nodes (n: $\#$ total nodes).

  So our goal is to get a $(1+\epsilon)$-approximation w.h.p.
  (ϵ is a very small constant, e.g., 0.01)

- Can we simply use sampling?

  No, it doesn't work. Consider the star, with degree sequence $(n-1, 1, 1, \ldots, 1)$.



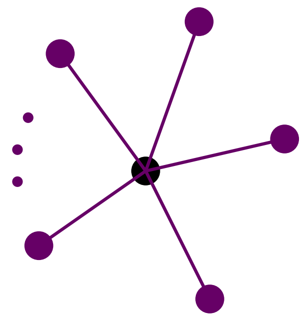
Why hard? You can't see everything in sublinear time!

- Computing exact average degree is impossible without querying at least $n - 1$ nodes (n: # total nodes).

  So our goal is to get a $(1 + \epsilon)$-approximation w.h.p.
  (ϵ is a very small constant, e.g., 0.01)

- Can we simply use sampling?

  No, it doesn't work. Consider the star, with degree sequence $(n - 1, 1, \ldots, 1)$.

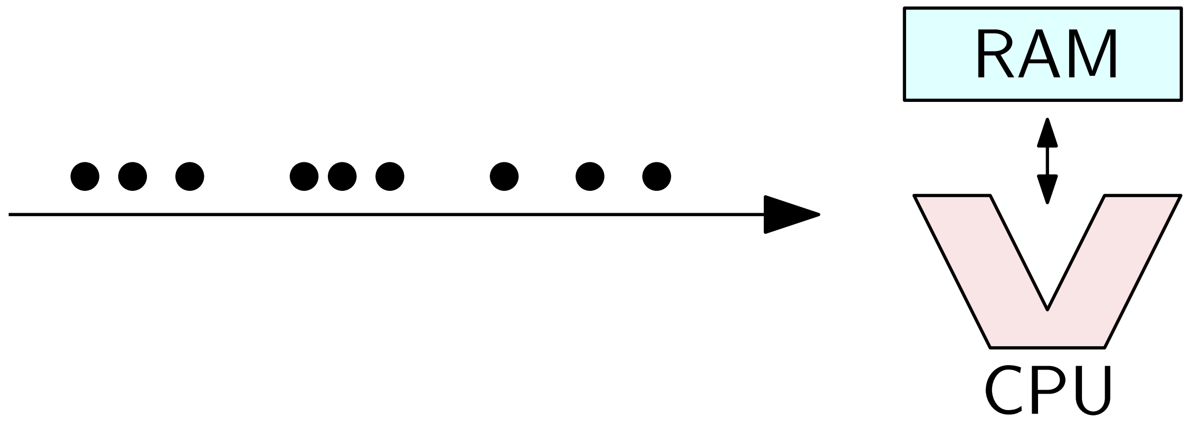


- So can we do anything non-trivial?

  (think about it, and we will discuss later in the course)

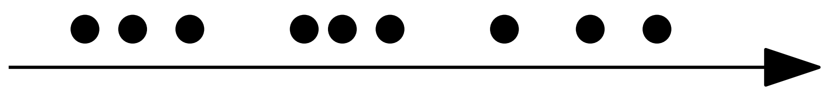
Sublinear in space

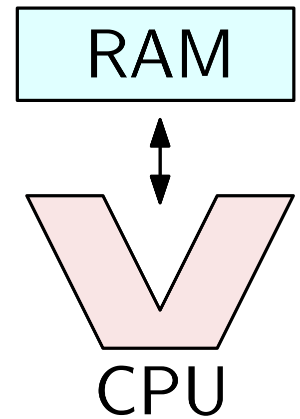
- **The data stream model** (Alon, Matias and Szegedy 1996)



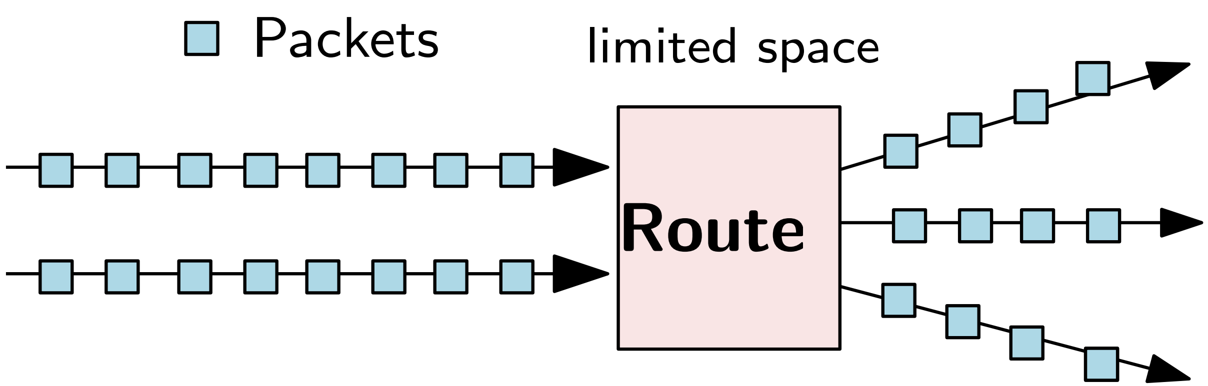
Sublinear in space

- **The data stream model** (Alon, Matias and Szegedy 1996)





- **Applications**
  - Internet Router.



The router wants to maintain some statistics on data.
E.g., want to detect anomalies for security.

  - Stock data, ad auction, flight logs on tapes, etc.

Why hard? You do see everything but then “forget”!

- **Game** 1: A sequence of numbers

Why hard? You do see everything but then “forget”!

- **Game** 1: A sequence of numbers

52

Why hard? You do see everything but then “forget”!

- **Game** 1: A sequence of numbers

45

Why hard? You do see everything but then “forget”!

- **Game** 1: A sequence of numbers

18

Why hard? You do see everything but then "forget"!

- **Game** 1: A sequence of numbers

23

Why hard? You do see everything but then "forget"!

- **Game** 1: A sequence of numbers

17

Why hard? You do see everything but then "forget"!

- **Game** 1: A sequence of numbers

41

Why hard? You do see everything but then “forget”!

- **Game** 1: A sequence of numbers

33

Why hard? You do see everything but then “forget”!

- **Game** 1: A sequence of numbers

29

Why hard? You do see everything but then “forget”!

- **Game** 1: A sequence of numbers

49

Why hard? You do see everything but then “forget”!

- **Game** 1: A sequence of numbers

12

Why hard? You do see everything but then “forget”!

- **Game** 1: A sequence of numbers

35

Why hard? You do see everything but then “forget”!

- **Game** 1: A sequence of numbers

  Q: What’s the median?

Why hard? You do see everything but then "forget"!

- **Game** 1: A sequence of numbers

  Q: What's the median?

  A: 33

Why hard? Cannot store everything.

- **Game** 1: A sequence of numbers

  Q: What's the median?

  A: 33

- **Game** 2: Relationships between
  Alice, Bob, Carol, Dave, Eva and Paul

Why hard? Cannot store everything.

- **Game** 1: A sequence of numbers

  Q: What's the median?

  A: 33

- **Game** 2: Relationships between
  Alice, Bob, Carol, Dave, Eva and Paul

  Alice and Bob become friends

Why hard? Cannot store everything.

- **Game** 1: A sequence of numbers

  Q: What's the median?

  A: 33

- **Game** 2: Relationships between
  Alice, Bob, Carol, Dave, Eva and Paul

  Carol and Eva become friends

Why hard? Cannot store everything.

- **Game** 1: A sequence of numbers

  Q: What's the median?

  A: 33

- **Game** 2: Relationships between
  Alice, Bob, Carol, Dave, Eva and Paul

  Eva and Bob become friends

Why hard? Cannot store everything.

- **Game** 1: A sequence of numbers

  Q: What's the median?

  A: 33

- **Game** 2: Relationships between
  Alice, Bob, Carol, Dave, Eva and Paul

  Dave and Paul become friends

Why hard? Cannot store everything.

- **Game** 1: A sequence of numbers

  Q: What's the median?

  A: 33

- **Game** 2: Relationships between Alice, Bob, Carol, Dave, Eva and Paul

  Alice and Paul become friends

Why hard? Cannot store everything.

- **Game** 1: A sequence of numbers

  Q: What's the median?

  A: 33

- **Game** 2: Relationships between
  Alice, Bob, Carol, Dave, Eva and Paul

  Eva and Bob unfriends

Why hard? Cannot store everything.

- **Game** 1: A sequence of numbers

  Q: What's the median?

  A: 33

- **Game** 2: Relationships between
  Alice, Bob, Carol, Dave, Eva and Paul

  Alice and Dave become friends

Why hard? Cannot store everything.

- **Game** 1: A sequence of numbers

  Q: What's the median?

  A: 33

- **Game** 2: Relationships between
  Alice, Bob, Carol, Dave, Eva and Paul

  Bob and Paul become friends

Why hard? Cannot store everything.

- **Game** 1: A sequence of numbers

  Q: What's the median?

  A: 33

- **Game** 2: Relationships between Alice, Bob, Carol, Dave, Eva and Paul

  Dave and Paul unfriends

Why hard? Cannot store everything.

- **Game** 1: A sequence of numbers

  Q: What's the median?

  A: 33

- **Game** 2: Relationships between Alice, Bob, Carol, Dave, Eva and Paul

  Dave and Carol become friends

Why hard? Cannot store everything.

- **Game** 1: A sequence of numbers

  Q: What's the median?

  A: 33

- **Game** 2: Relationships between
  Alice, Bob, Carol, Dave, Eva and Paul

  Q: Are Eva and Bob connected by friends?

Why hard? Cannot store everything.

- **Game** 1: A sequence of numbers

  Q: What's the median?

  A: 33

- **Game** 2: Relationships between
  Alice, Bob, Carol, Dave, Eva and Paul

  Q: Are Eva and Bob connected by friends?

  A: YES. Eva $\Leftrightarrow$ Carol $\Leftrightarrow$ Dave $\Leftrightarrow$ Alice $\Leftrightarrow$ Bob

Why hard? Cannot store everything.

- **Game** 1: A sequence of numbers

  Q: What's the median?

  A: 33

- **Game** 2: Relationships between
  Alice, Bob, Carol, Dave, Eva and Paul

  Q: Are Eva and Bob connected by friends?

  A: YES. Eva $\Leftrightarrow$ Carol $\Leftrightarrow$ Dave $\Leftrightarrow$ Alice $\Leftrightarrow$ Bob

- Have to allow approx/randomization given a small memory.

Sublinear in communication

- **The model** $x_1 = 010011$ $x_2 = 111011$

$x_3 = 111111$

$x_k = 100011$

They want to jointly compute $f(x_1, x_2, \ldots, x_k)$ (e.g., f is $\#$ distinct ele)

Goal: minimize total bits of communication

Sublinear in communication

- **The model**

$x_1 = 010011$ $x_2 = 111011$

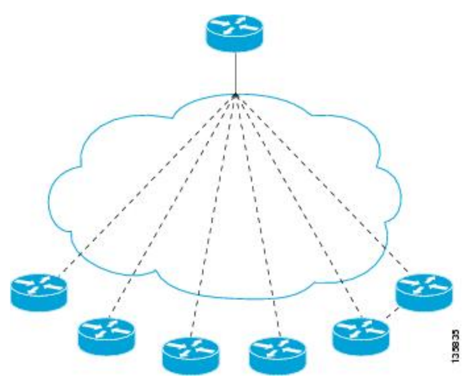
$x_3 = 111111$

$x_k = 100011$

They want to jointly compute $f(x_1, x_2, \ldots, x_k)$ (e.g., f is # distinct ele)

Goal: minimize total bits of communication

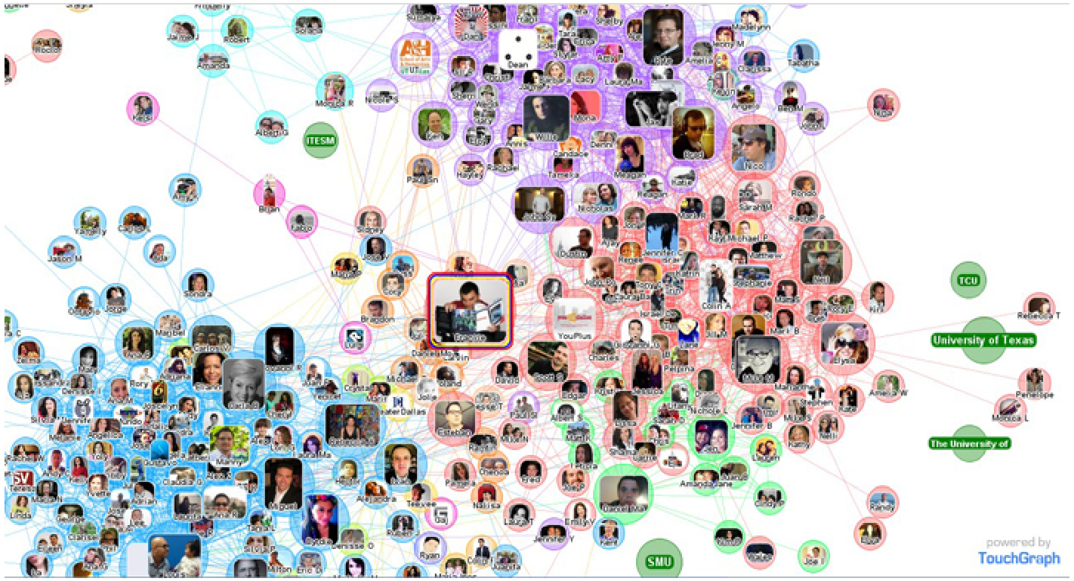
- **Applicaitons**



Why hard? You do not have a global view of the data.

Let's think about the **graph connectivity** problem:
k machine each holds a set of edges of a graph.

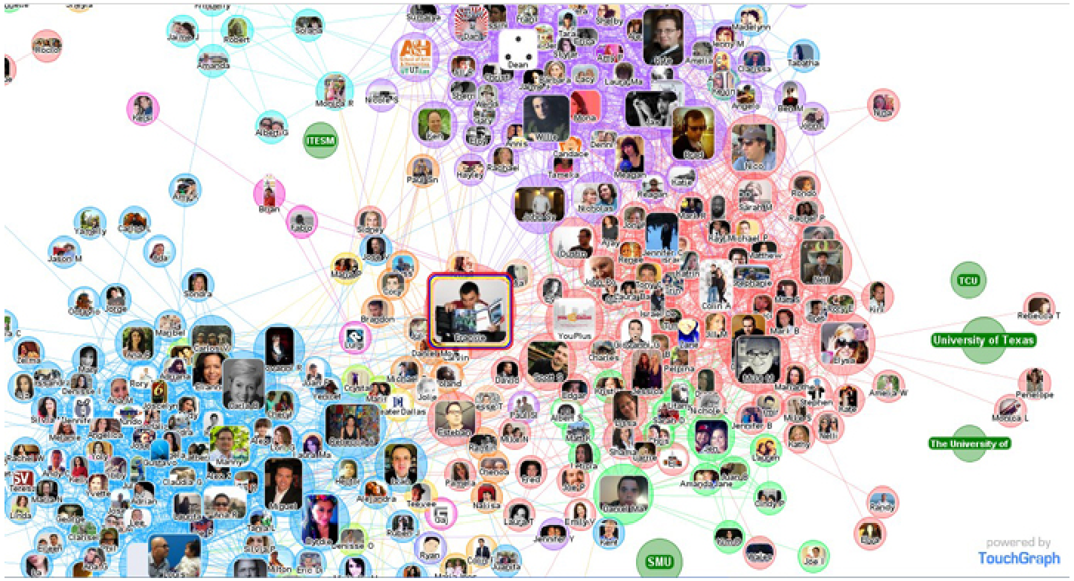
Goal: compute whether the graph is connected.



Why hard? You do not have a global view of the data.

Let's think about the **graph connectivity** problem:
k machine each holds a set of edges of a graph.

Goal: compute whether the graph is connected.

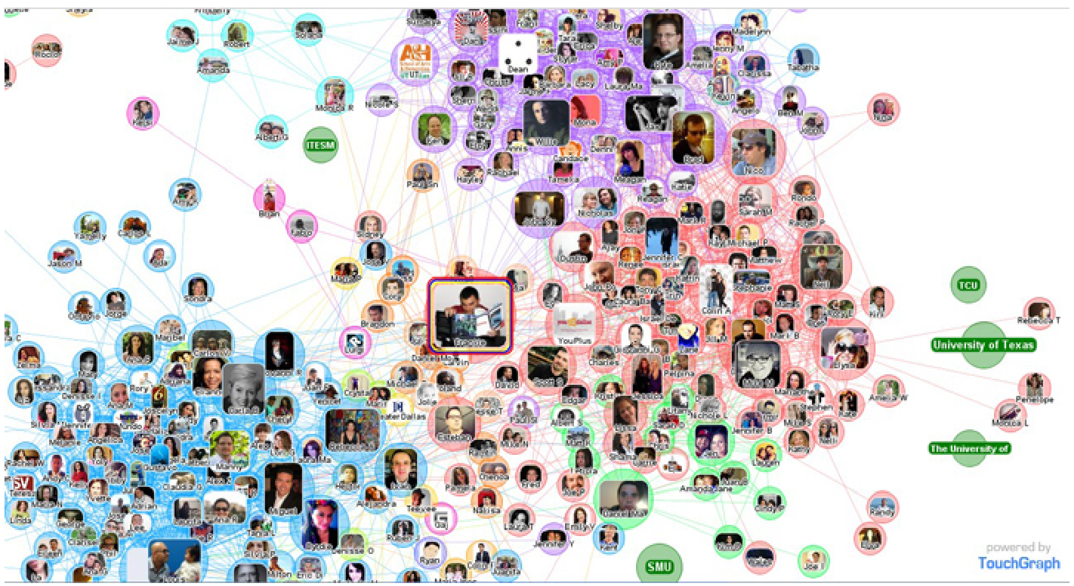


A trivial solution: each machine sends a local spanning forest to the first machine. Cost $O(kn \log n)$ bits.

Why hard? You do not have a global view of the data.

Let's think about the **graph connectivity** problem:
k machine each holds a set of edges of a graph.

Goal: compute whether the graph is connected.



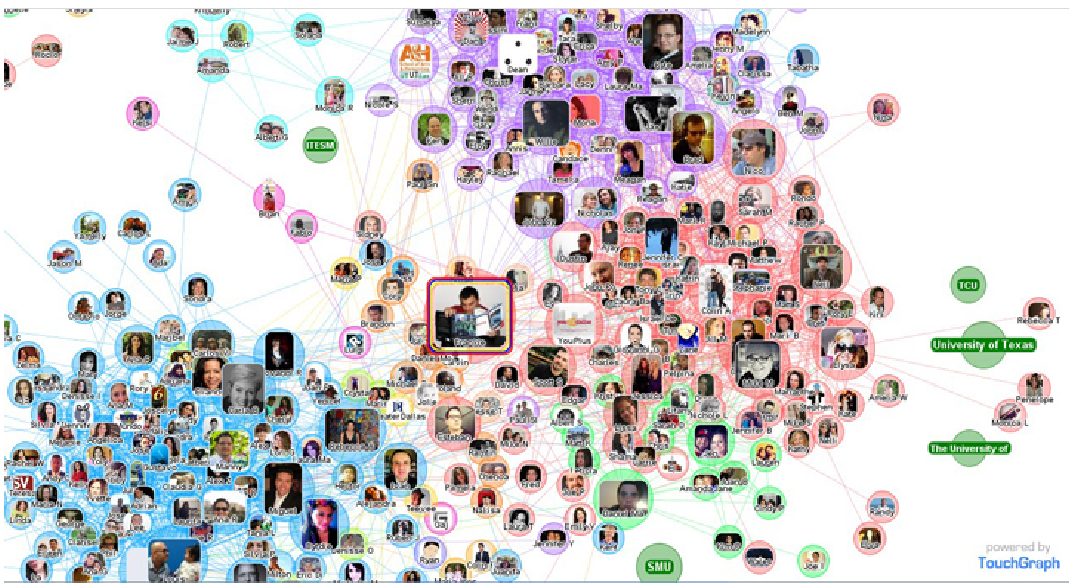
A trivial solution: each machine sends a local spanning forest to the first machine. Cost $O(kn \log n)$ bits.

Can we do better, e.g., $o(kn)$ bits of communication?

Why hard? You do not have a global view of the data.

Let's think about the **graph connectivity** problem:
k machine each holds a set of edges of a graph.

Goal: compute whether the graph is connected.



A trivial solution: each machine sends a local spanning forest to the first machine. Cost $O(kn \log n)$ bits.

Can we do better, e.g., $o(kn)$ bits of communication?

What if the graph is *node partitioned* among the k machines?
That is, each node is stored in 1 machine with all adjancent edges.

Problems

Statistical problems

- Frequency moments F_p

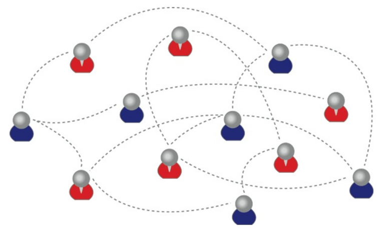
  F_0: #distinct elements

  F_2: size of self-join
- Heavy hitters
- Quantile
- Entropy
- ...

Problems

Statistical problems

- Frequency moments F_p
  F_0: #distinct elements
  F_2: size of self-join
- Heavy hitters
- Quantile
- Entropy
- ...



Graph problems

- Connectivity
- Bipartiteness
- Counting triangles
- Matching
- Minimum spanning tree
- ...

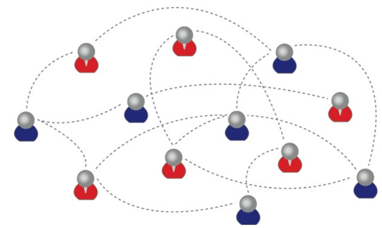
Problems

Statistical problems

- Frequency moments F_p

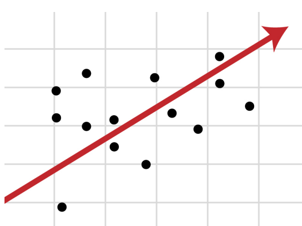
  F_0: #distinct elements

  F_2: size of self-join
- Heavy hitters
- Quantile
- Entropy
- ...



Graph problems

- Connectivity
- Bipartiteness
- Counting triangles
- Matching
- Minimum spanning tree
- ...



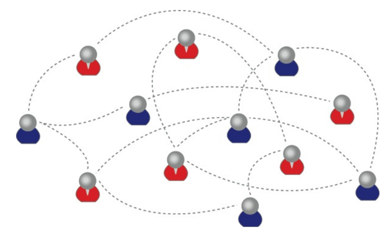
Numerical linear algebra

- L_p regression
- Low-rank approximation
- ...

Problems

Statistical problems

- Frequency moments F_p
  F_0: #distinct elements
  F_2: size of self-join
- Heavy hitters
- Quantile
- Entropy
- . . .



Graph problems

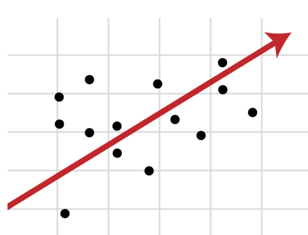
- Connectivity
- Bipartiteness
- Counting triangles
- Matching
- Minimum spanning tree
- . . .

DB queries

- Conjuntive queries

Strings

- Edit distance
- Longest increasing sequence



Numerical linear algebra

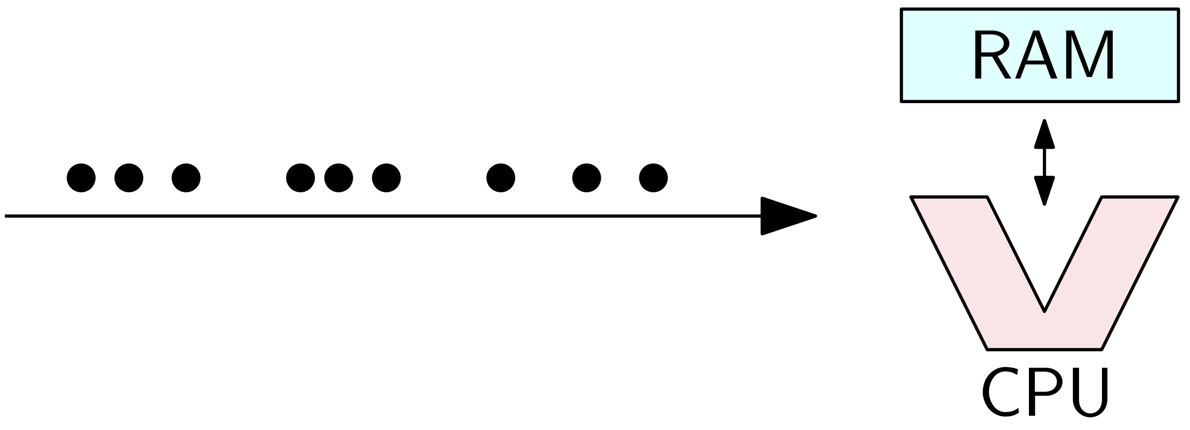
- L_p regression
- Low-rank approximation
- . . .

Geometry problems

- Clustering
- Earth-Mover Distance

. . .

Example: random sampling in data stream



A toy example: Reservoir Sampling

Tasks: Find a **uniform sample** from a stream of unknown length, can we do it in $O(1)$ space?

A toy example: Reservoir Sampling

Tasks: Find a **uniform sample** from a stream of unknown length, can we do it in $O(1)$ space?

Algorithm: Store 1-st item. When the i-th ($i > 1$) item arrives

With probability $1/i$, replace the current sample;
With probability $1 - 1/i$, throw it away.

A toy example: Reservoir Sampling

Tasks: Find a **uniform sample** from a stream of unknown length, can we do it in $O(1)$ space?

Algorithm: Store 1-st item. When the i-th $(i > 1)$ item arrives

With probability $1/i$, replace the current sample;
With probability $1 - 1/i$, throw it away.

Correctness: each item is included in the final sample w.p.
$\frac{1}{i} \times (1 - \frac{1}{i+1}) \times \ldots \times (1 - \frac{1}{n}) = \frac{1}{n}$ (n: total # items)

Space: $O(1)$

Maintain a sample for Sliding Windows

Tasks: Find a uniform sample from the **last** w **items**.

Maintain a sample for Sliding Windows

Tasks: Find a uniform sample from the **last** w **items**.

Algorithm:

– For each x_i, we pick a random value $v_i \in (0, 1)$.
– In a window $< x_{j-w+1}, \ldots, x_j >$, return value x_i with smallest v_i.
– To do this, maintain the set of all x_i in sliding window whose v_i value is minimal among subsequent values.

Maintain a sample for Sliding Windows

Tasks: Find a uniform sample from the **last** w **items**.

Algorithm:

– For each x_i, we pick a random value $v_i \in (0, 1)$.
– In a window $< x_{j-w+1}, \ldots, x_j >$, return value x_i with smallest v_i.
– To do this, maintain the set of all x_i in sliding window whose v_i value is minimal among subsequent values.

Correctness: Obvious.

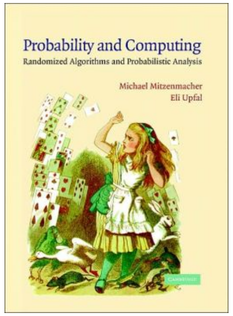
Space (expected): $1/w + 1/(w-1) + \ldots + 1/1 = \log w$.

Resources

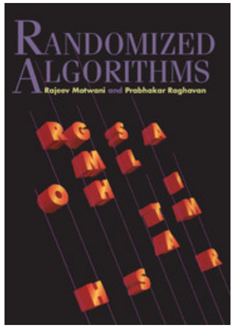
- There is no textbook for the class.

  Reference for part of the course: lecture notes by Amit Chakrabarti

- Background on Randomized Algorithms:
  - Probability and Computing
    by Mitzenmacher and Upfal
    (Advanced undergraduate textbook)



  - Randomized Algorithms
    by Motwani and Raghavan
    (Graduate textbook)



Instructors

- Instructor: Qin Zhang
  Email: qzhangcs@iu.edu
  Office hours: by appointment

Grading

I am thinking about it. Assignments + Final Project

Prerequisites

A research-oriented course. Will be quite mathematical.

One is expected to know:
basics on algorithm design and analysis + basic probability.

e.g., have taken B403 "Introduction to Algorithm Design and Analysis" or equivalent courses.

I will NOT start with things like big-O notations, the definitions of expectation and variance, and hashing.

Thank you!