Privacy Preservation in the Context of Big Data Processing

Kavé Salamatian Universite de Savoie
Eiko Yoneki University of Cambridge

- PART I – Eiko Yoneki

Large-Scale Data Processing

- PART II - Kavé Salamatian

Big Data and Privacy

2

PART I: Large-Scale Data Processing

Eiko Yoneki

eiko.yoneki@cl.cam.ac.uk
http://www.cl.cam.ac.uk/~ey204

Systems Research Group
University of Cambridge Computer Laboratory

Outline

- **What and Why large data?**
- Technologies
- Analytics
- Applications
- Privacy → Kavé

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Source of Big Data

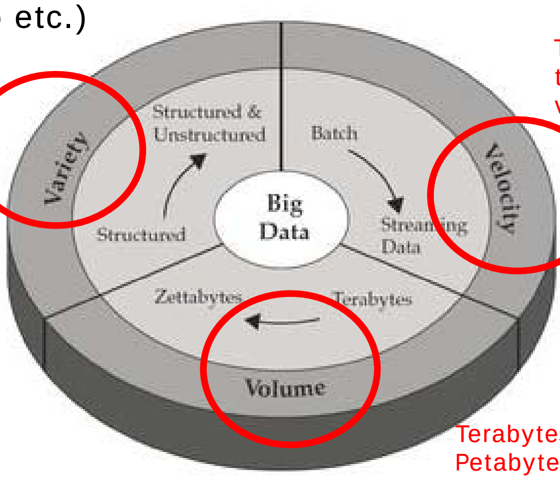
- Facebook:
  - 40+ billion photos (100PB)
  - 6 billion messages per day (5-10 TB)
  - 900 million users (1 trillion connections?)
- Common Crawl:
  - Covers 5 million web pages
  - 50 TB data
- Twitter Firehose:
  - 350 million tweet/day x 2-3Kb/tweet ~ 1TB/day
- CERN
  - 15 PB/year - Stored in RDB
- Google:
  - 20PB a day (2008)
- ebay
  - 9PB of user data+ >50 TB/day
- US census data
  - Detailed demographic data
- Amazon web services
  - S3 450B objects, peak 290K request/sec
- JPMorganChase
  - 150PB on 50K+ servers with 15K apps running

5

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

3Vs of Big Data

- Volume: terabytes even petabytes scale
- Velocity: Time sensitive – streaming
- Variety: beyond structured data (e.g. text, audio, video etc.)



Time-sensitive to maximise its value

Beyond structured data

Terabytes or even Petabytes scale

6

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Significant Financial Value

US health care
- $300 billion value per year
- ~0.7 percent annual productivity growth

Europe public sector administration
- €250 billion value per year
- ~0.5 percent annual productivity growth

Global personal location data
- $100 billion+ revenue for service providers
- Up to $700 billion value to end users

US retail
- 60+% increase in net margin possible
- 0.5–1.0 percent annual productivity growth

Manufacturing
- Up to 50 percent decrease in product development, assembly costs
- Up to 7 percent reduction in working capital

SOURCE: McKinsey Global Institute analysis

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Gnip

- Grand central station for Social Web Stream
- Aggregate several TB of new social data daily

Climate Corporation

- 14TB of historical weather data
- 30 technical staff including 12 PhDs
- 10,000 sales agents

9

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

FICO

- 50+ years of experience doing credit ratings
- Transitioning to predictive analytics

10

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Why Big Data?

- Hardware and software technologies can manage ocean of data
- Increase of Storage Capacity
- Increase of Processing Capacity
- Availability of Data

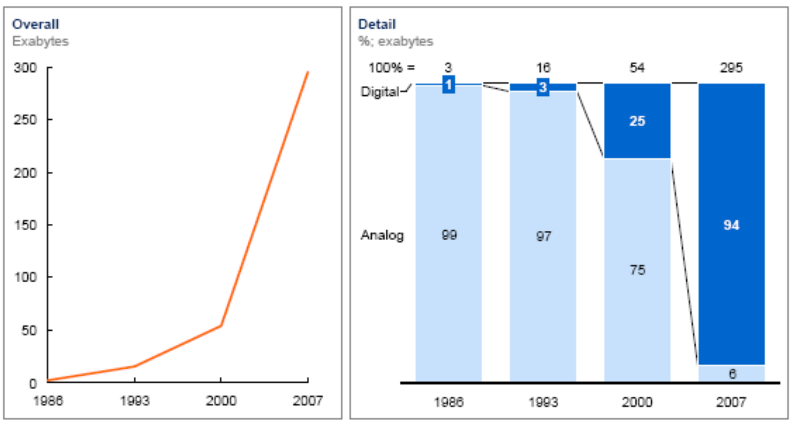
11

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Data Storage

Data storage has grown significantly, shifting markedly from analog to digital after 2000

Global installed, optimally compressed, storage



NOTE: Numbers may not sum due to rounding.

SOURCE: Hilbert and López, "The world's technological capacity to store, communicate, and compute information," *Science*, 2011

12

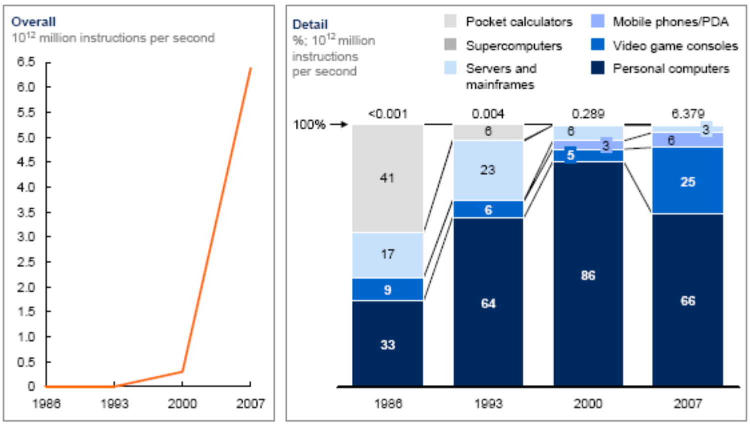
SOURCE: McKinsey Global Institute analysis

UNIVERSITY OF CAMBRIDGE

Computation Capability

Computation capacity has also risen sharply

Global installed computation to handle information



NOTE: Numbers may not sum due to rounding.

SOURCE: Hilbert and López, "The world's technological capacity to store, communicate, and compute information," *Science*, 2011

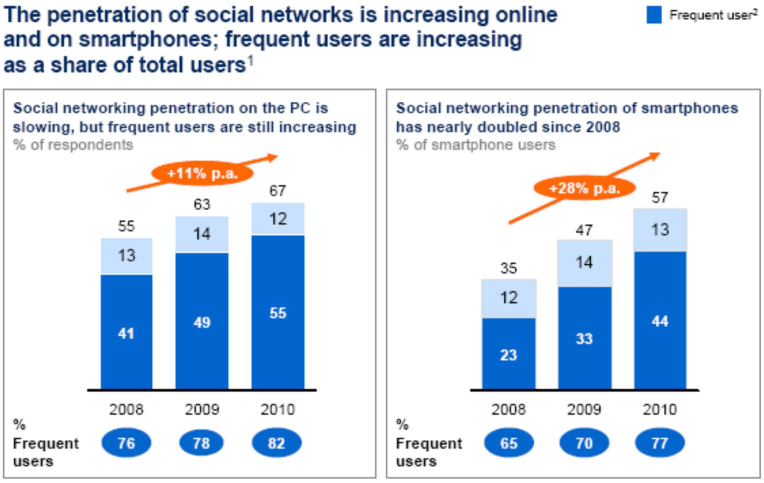
13

SOURCE: McKinsey Global Institute analysis

UNIVERSITY OF CAMBRIDGE

Data from Social networks

The penetration of social networks is increasing online and on smartphones; frequent users are increasing as a share of total users[1]



1 Based on penetration of users who browse social network sites. For consistency, we exclude Twitter-specific questions (added to survey in 2009) and location-based mobile social networks (e.g., Foursquare, added to survey in 2010).

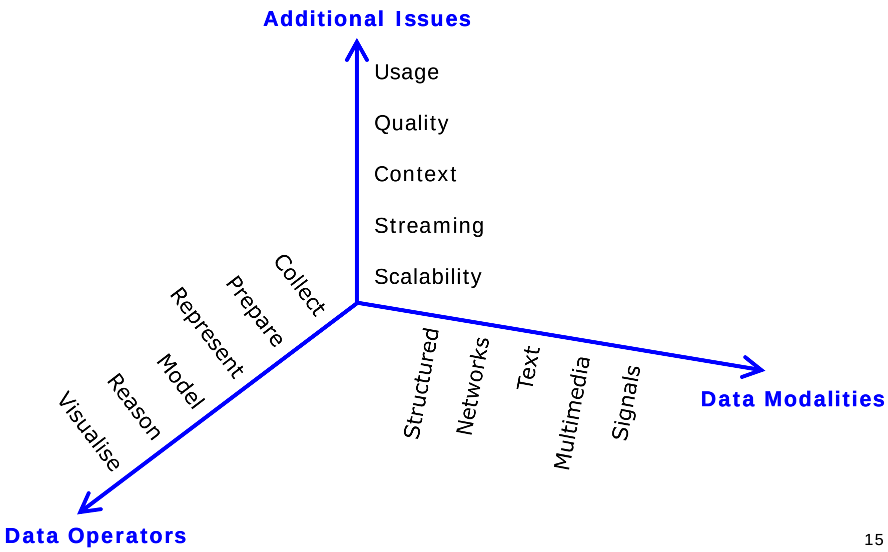
2 Frequent users defined as those that use social networking at least once a week.

SOURCE: McKinsey iConsumer Survey

14

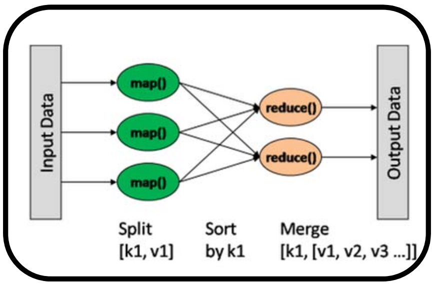
SOURCE: McKinsey Global Institute analysis

Issues to Process Big Data



Outline

- What and Why large data?
- **Technologies**
- Analytics
- Applications
- Privacy



Techniques for Analysis

- Applying these techniques: larger and more diverse datasets can be used to generate more numerous and insightful results than smaller, less diverse ones

- Classification
- Cluster analysis
- Crowd sourcing
- Data fusion/integration
- Data mining
- Ensemble learning
- Genetic algorithms
- Machine learning
- NLP
- Neural networks
- Network analysis
- Optimisation
- Pattern recognition
- Predictive modelling
- Regression
- Sentiment analysis
- Signal processing
- Spatial analysis
- Statistics
- Supervised learning
- Simulation
- Time series analysis
- Unsupervised learning
- Visualisation

17

Technologies for Big Data

- Distributed systems
  - Cloud (e.g. Amazon EC2 - Infrastructure as a service)
- Storage
  - Distributed storage (e.g. Amazon S3)
- Programming model
  - Distributed processing (e.g. MapReduce)
- Data model/indexing
  - High-performance schema-free database (e.g. NoSQL DB)
- Operations on big data
  - Analytics – Realtime Analytics

18

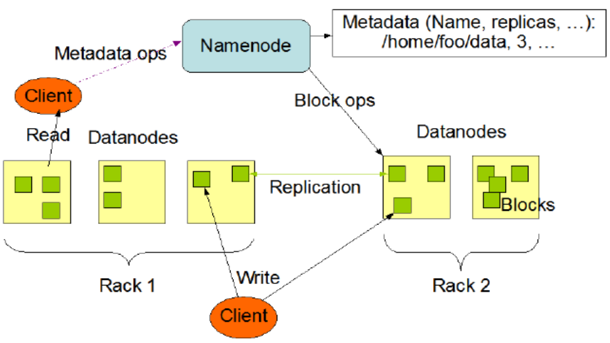
Distributed Infrastructure

- Computing + Storage transparently
  - Cloud computing, Web 2.0
  - Scalability and fault tolerance
- Distributed servers
  - Amazon EC2, Google App Engine, Elastic, Azure
  - E.g. EC2:
    - Pricing? Reserved, on-demand, spot, geography
    - System? OS, customisations
    - Sizing? RAM/CPU based on tiered model
    - Storage? Quantity, type
    - Networking / security
- Distributed storage
  - Amazon S3
  - Hadoop Distributed File System (HDFS)
  - Google File System (GFS) - BigTable
  - Hbase

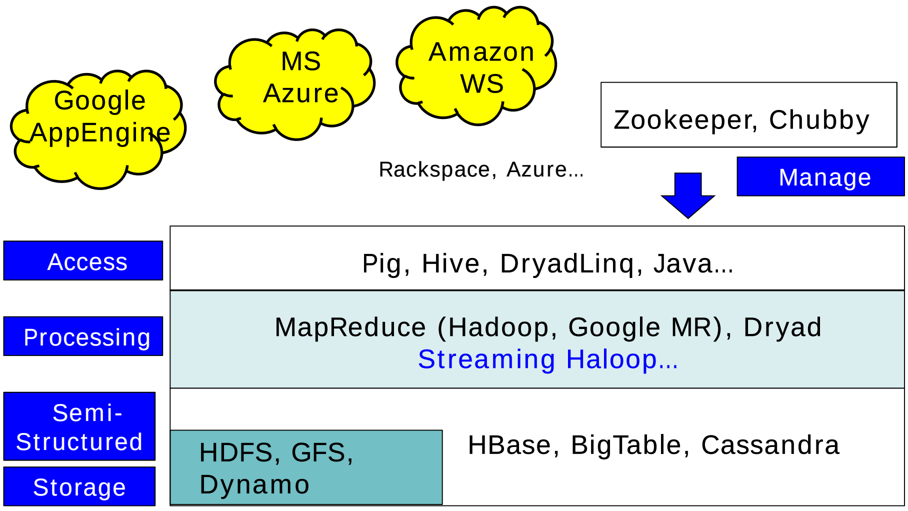
Distributed Storage

- E.g. Hadoop Distributed File System (HDFS)
  - High performance distributed file system
  - Asynchronous replication
  - Data divided into 64/128 MB blocks (replicated 3 times)
  - NameNode holds file system metadata
  - Files are broken up and spread over DataNodes

HDFS Architecture



Distributed Infrastructure

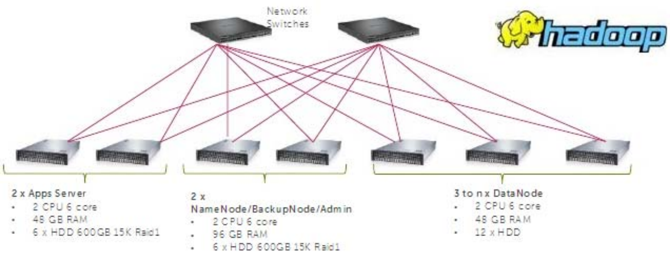


Challenges

- Big data→ to scale and build on distribution and combine theoretically unlimited number of machines to one single distributed storage
- Distribute and shard parts over many machines
  - Still fast traversal and read to keep related data together
  - Data store including NoSQL
  - Scale out instead scale up
- Avoid naïve hashing for sharding
  - Do not depend of the number of nodes
  - Difficult add/remove nodes
  - Trade off – data locality, consistency, availability, read/write/search speed, latency etc.
- Analytics requires both real time and post fact analytics

Hadoop

- Founded in 2004 by a Yahoo! Employee
- Spun into open source Apache project
- General purpose framework for Big Data
  - MapReduce implementation
  - Support tools (e.g. distributed storage, concurrency)
  - Use by everybody...(Yahoo!, Facebook, Amazon, MS, Apple)



23

Amazon Web Services

- Launched 2006
- Largest most popular cloud computing platform
- Elastic Compute Cloud (EC2)
  - Rent Elastic compute units by the hour: one 1 GH machine
  - Can choose Linux, FreeBSD, Solaris, and Windows
  - Virtual private servers running on Xen
  - Pricing: US$0.02 – 2.50 per hour
- Simple Storage Service (S3)
  - Index by bucket and key
  - Accessible via HTTP, SOAP and BitTorrent
  - Over 1 trillion objects now uploaded
  - Pricing: US$0.05-0.10 per GB per month
- Stream Processing Service (S4)
- Other AWS:
  - Elastic MapReduce (Hadoop on EC2 with S3)
  - SQL Database
  - Content delivery networks, caching

24

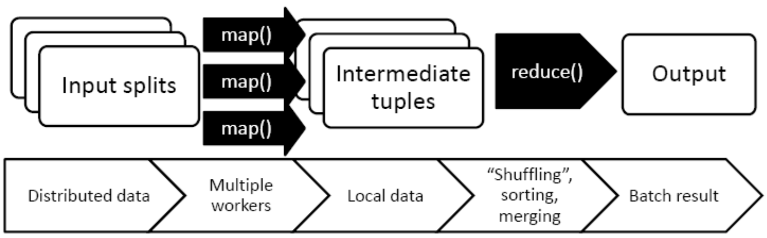
Distributed Processing

- Non standard programming models
  - Use of cluster computing
  - No traditional parallel programming models (e.g. MPI)
  - New model: e.g. MapReduce

25

MapReduce

- Target problem needs to be parallelisable
- Split into a set of smaller code (map)
- Next small piece of code executed in parallel
- Finally a set of results from map operation get synthesised into a result of the original problem (reduce)



26

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Task Coordination

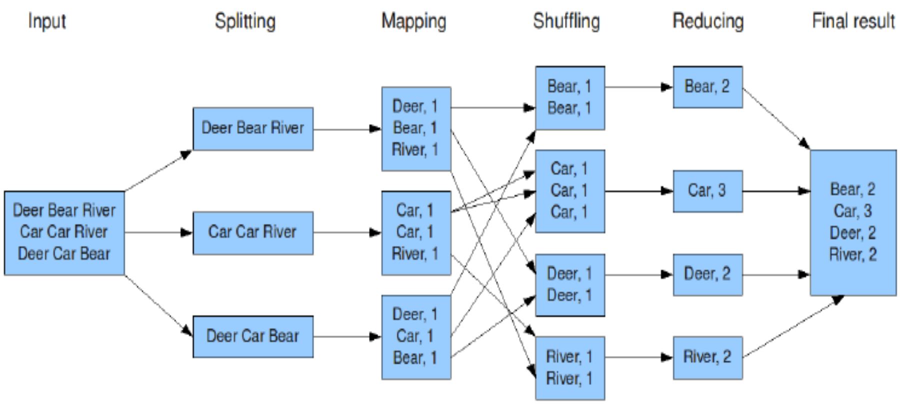
- Typical architecture utilises a single master and multiple (unreliable) workers
- Master holds state of current configuration, detects node failure, and schedules work based on multiple heuristics. Also coordinates resources between multiple jobs
- Workers perform work! Both mapping and reducing, possibly at the same time

27

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Example: Word Count

The overall MapReduce word count process



28

Example: Word Count

```
public class MapClass extends MapReduceBase implements
Mapper<LongWritable, Text, Text, IntWritable> {
      private final static IntWritable one = new IntWritable(1);
      private Text word = new Text();
      public void map(LongWritable key, Text value,
                      OutputCollector<Text, IntWritable> output,
                      Reporter reporter) throws IOException {
          String line = value.toString();
          StringTokenizer itr = new StringTokenizer(line);
          while (itr.hasMoreTokens()) {
              word.set(itr.nextToken());
              output.collect(word, one);
          }
      }
}
```

```
public class Reduce extends MapReduceBase implements
Reducer<Text, IntWritable, Text, IntWritable> {

    public void reduce(Text key, Iterator<IntWritable> values,
                       OutputCollector<Text, IntWritable>
output,
                       Reporter reporter) throws IOException {
        int sum = 0;
        while (values.hasNext()) {
            sum += values.next().get();
        }
        output.collect(key, new IntWritable(sum));
    }

}
```


Example: Word Count

```
public class WordCount {
    public static void main(String[] args) throws Exception {
        JobConf conf = new JobConf(WordCount.class);
        conf.setJobName("wordcount");

        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(IntWritable.class);

        conf.setMapperClass(Map.class);
        conf.setCombinerClass(Reduce.class);
        conf.setReducerClass(Reduce.class);

        conf.setInputFormat(TextInputFormat.class);
        conf.setOutputFormat(TextOutputFormat.class);

        FileInputFormat.setInputPaths(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));
        JobClient.runJob(conf);
    }
}
```

Example from rapidgremlin.com

CIEL: Dynamic Task Graphs

- MapReduce prescribes a **task graph** that can be adapted to many problems
- Later execution engines such as Dryad allow more flexibility, for example to combine the results of multiple separate computations
- CIEL takes this a step further by allowing the task graph to be specified at run time – for example:

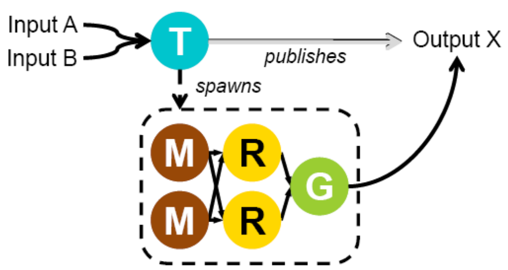
  while (!converged) spawn(tasks);

Tutorial: http://www.cambridgeplus.net/tutorials/CIEL-DCN/

31

Dynamic Task Graph

- Data-dependent control flow



- CIEL: Execution engine for dynamic task graphs

  (D. Murray et al. CIEL: a universal execution engine for distributed data-flow computing, NSDI 2011)

32

Data Model/Indexing

- Support large data
- Fast and flexible
- Operate on distributed infrastructure

- Is SQL Database sufficient?

33

Traditional SQL Databases

```
TABLE instructor
+-------+----------------+
|  ID   | Name           |
+-------+----------------+
|  14   | David Singleton |
|  27   | Joseph Bonneau |
|  52   | Pete Warden    |
+-------+----------------+

TABLE lectures

+-------+-------------------+-----------+
|  ID   | Title             |  Lecturer |
+-------+-------------------+-----------+
|  1    | BD at Google      |  14       |
|  2    | Overview of BD    |  27       |
|  3    | Algorithms for BD |  27       |
|  4    | BD at startups    |  14       |
+-------+-------------------+-----------+
```

Most interesting queries require computing **joins**

34

NoSQL (Schema Free) Database

- NoSQL database
  - Support large data
  - Operate on distributed infrastructure (e.g. Hadoop)
  - Based on key-value pairs (no predefined schema)
  - Fast and flexible
- Pros: Scalable and fast
- Cons: Fewer consistency/concurrency guarantees and weaker queries support
- Implementations
  - MongoDB
  - CouchDB
  - Cassandra
  - Redis
  - BigTable
  - Hibase
  - Hypertable
  - ...

35

Data Assumptions

Traditional RDBMS (SQL)	NoSQL
integrity is mission-critical	OK as long as most data is correct
data format consistent, well-defined	data format unknown or inconsistent
data is of long-term value	data will be replaced
data updates are frequent	write-once, ready multiple
predictable, linear growth	unpredictable growth (exponential?)
non-programmers writing queries	only programmers writing queries
regular backup	replication
access through master server	sharding

36

NoSQL Database

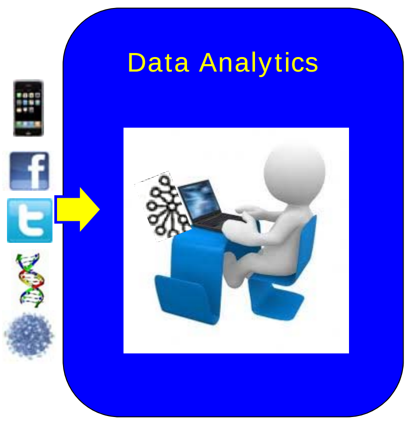
- Maintain unique keys per row
- Complicated multi-valued columns for rich query

RowKey	TimeStamp	ColumnFamily contents	ColumnFamily anchor
com.cnn.www	t1	contents.html = ...	anchor:cnnsi.com = "CNN"
com.cnn.www	t0	contents.html = ...	anchor:cnnsi.com = "News"
...	...	...	...
uk.ac.cam.www	t1	contents.html = ...	anchor:cl.cam.ac.uk = "Home"
uk.ac.cam.cl.www	t1	contents.html = ...	anchor:cl.cam.ac.uk/jcb82 = "My Lab" anchor:cam.ac.uk = "Computer Lab"

37

Outline

- What and Why large data?
- Technologies
- **Analytics**
- Applications
- Privacy



38

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Do we need new Algorithms?

- Can't always store all data
  - Online/streaming algorithms
- Memory vs. disk becomes critical
  - Algorithms with limited passes
- N^2 is impossible
  - Approximate algorithms
- Data has different relation to various other data
  - Algorithms for high-dimensional data

39

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Complex Issues with Big Data

- Because of large amount of data, statistical analysis might produce meaningless result
- Example:
  - We want to find (unrelated) people who at least twice have stayed at the same hotel on the same day
    - 10^9 people being tracked.
    - 1000 days.
    - Each person stays in a hotel 1% of the time (1 day out of 100)
    - Hotels hold 100 people (so 10^5 hotels).
    - If everyone behaves randomly (i.e., no terrorists) will the data mining detect anything suspicious?
  - Expected number of "suspicious" pairs of people:
    - 250,000
    - … too many combinations to check – we need to have some additional evidence to find "suspicious" pairs of people in some more efficient way

Example taken from: Rajaraman, Ullman: Mining of Massive Datasets

40

Easy Cases

- Sorting
  - Google 1 trillion items (1PB) sorted in 6 Hours
- Searching
  - Hashing and distributed search

→ Random split of data to feed M/R operation

- But not all algorithms are parallelisable

41

More Complex Case: Stream Data

- Have we seen x before?
- Rolling average of previous K items
  - Sliding window of traffic volume
- Hot list – most frequent items seen so far
  - Probability start tracking new item
- Querying data streams
  - Continuous Query

42

UNIVERSITY OF CAMBRIDGE Computer Laboratory

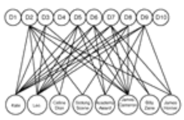
Typical Operation with Big Data

- Smart sampling of data
  - Reducing original data with maintaining statistical properties
- Find similar items → efficient multidimensional indexing
- Incremental updating of models → support streaming
- Distributed linear algebra → dealing with large sparse matrices
- Plus usual data mining, machine learning and statistics
  - Supervised (e.g. classification, regression)
  - Non-supervised (e.g. clustering..)

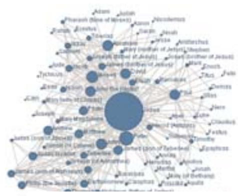
43

UNIVERSITY OF CAMBRIDGE Computer Laboratory

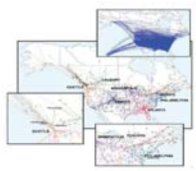
How about Graph Data



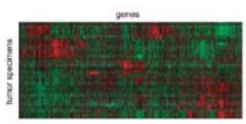
Bipartite graph of appearing phrases in documents



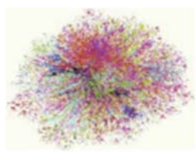
Social Networks



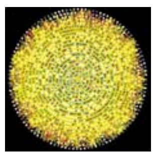
Airline Graph



Gene expression data



Internet Map [lumeta.com]



Protein Interactions [genomebiology.com]

44

Different Algorithms for Graph

- Different Algorithms perform differently

BFS
DFS
CC
SCC
SSSP
ASP
MIS
A*
Community
Centrality
Diameter
Page Rank
...

Algorithm	SQL Server PDW	DryadLINQ	SHS
PageRank	8,970	4,513	90,942
SALSA	2,034	439	163
SCC	475	446	214,858/1,073
WCC	4,207	3,844	1,976
ASP	30,379	17,089	246,944

Running time in seconds processing the graph with 50million English web pages with 16 servers (from Najork et al WSDM 2012)

45

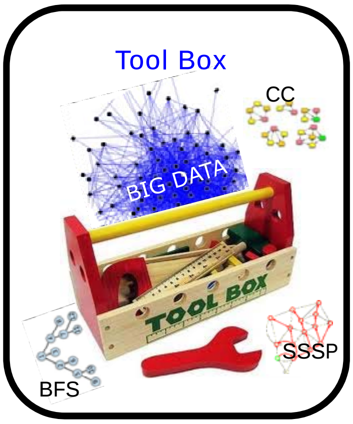
How to Process Big Graph Data?

- Data-Parallel (e.g. MapReduce)
  - Large datasets are partitioned across machines and replicated
  - No efficient random access to data
  - Graph algorithms are not fully parallelisable
- Parallel DB
  - Tabular format providing ACID properties
  - Allow data to be partitioned and processed in parallel
  - Graph does not map well to tabular format
- Moden NoSQL
  - Allow flexible structure (e.g. graph)
  - Trinity, Neo4J
  - In-memory graph store for improving latency (e.g. Redis, Scalable Hyperlink Store (SHS)) → expensive for petabyte scale workload

46

Big Graph Data Processing

- MapReduce is not suited for graph processing
  - Many iterations are needed for parallel graph processing
  - Intermediate results at every MapReduce iteration harm performance
- Graph specific data parallel
  - Multiple iterations needed to explore entire graph
  - Iterative algorithms common in Machine Learning, graph analysis

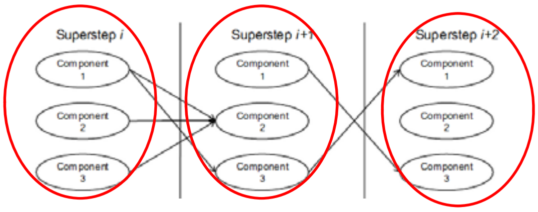


Data Parallel with Graph is Hard

- Designing Efficient Parallel Algorithms
  - Avoid Deadlocks on Access to Data
  - Prevent Parallel Memory Bottlenecks
  - Requires Efficient Algorithms for Data Parallel
- High Level Abstraction Helps → MapReduce
  - But processing millions of data with interdependent computation, difficult to deploy
- Data Dependency and Iterative Operation is Key
  - CIEL
  - GraphLab
- Graph Specific Data Parallel
  - Use of Bulk Synchronous Parallel Model
  - BSP enables peers to communicate only necessary data while data preserve locality

Bulk Synchronous Parallel Model

- Computation is sequence of iterations
- Each iteration is called a super-step
- Computation at each vertex in parallel



- Google Pregel: Vertex-based graph processing; defining a model based on computing locally at each vertex and communicating via message passing over vertex's available edges
  - BSP-based: Giraph, HAMA, GoldenORB

49

BSP Example

- Finding the largest value in a strongly connected graph

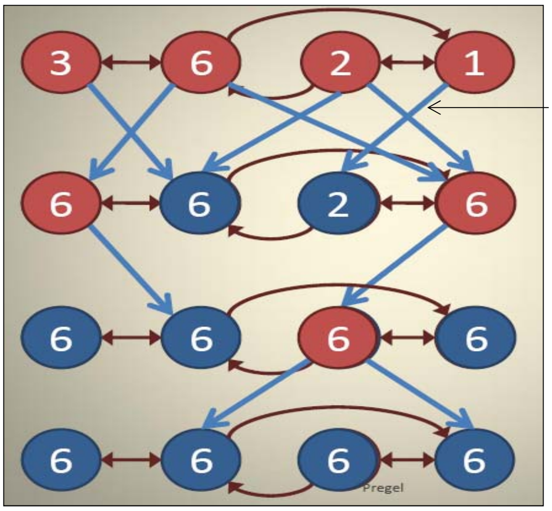
Local Computation

Communication

Local Computation

Communication

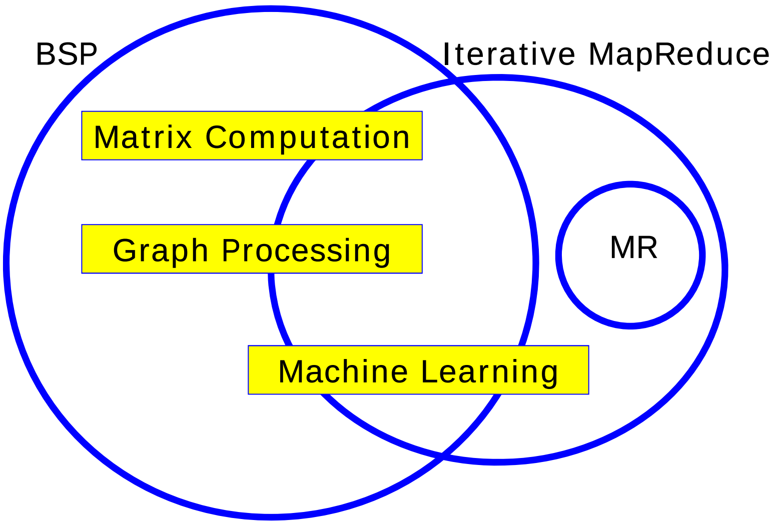
...



Message

50

Graph, Matrix, Machine Learning



51

Further Issues on Graph Processing

- Lot of work on computation
- Little attention to storage
  - Store LARGE amount of graph structure data (edge lists)
  - Efficiently move it to computation (algorithm)

Potential solutions:

- Cost effective but efficient storage
  - Move to SSDs from RAM
- Reduce latency
  - Blocking to improve spatial locality
  - Runtime prefetching
- Reduce storage requirements
  - Compressed Adjacency Lists

52

Outline

- What and Why large data?
- Technologies
- Analytics
- **Applications**
- Privacy

53

Applications

- Digital marketing Optimisation (e.g. web analytics)
- Data exploration and discovery (e.g. data science, new markets)
- Fraud detection and prevention (e.g. site integrity)
- Social network and relationship analysis (e.g. influence marketing)
- Machine generated data analysis (e.g. remote sensing)
- Data retention (i.e. data archiving)

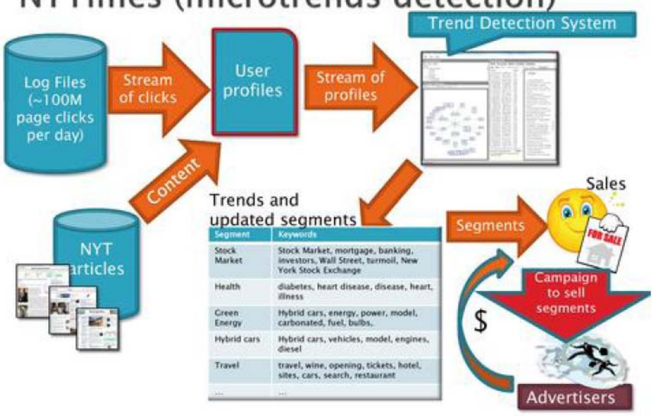
54

Recommendation

- Good recommendations can make a big difference when keeping a user on a web site
  - ...the key is how rich the context model a system is using to select information for a user
  - Bad recommendations <1% users, good ones >5% users click
  - 200clicks/sec

Online Advertisement

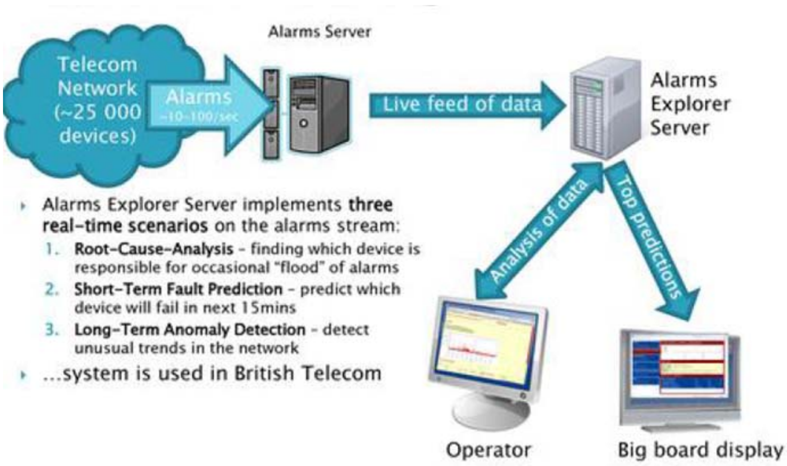
Application: Online Advertising for NYTimes (microtrends detection)



Segment	Keywords
Stock Market	Stock Market, mortgage, banking, investors, Wall Street, turmoil, New York Stock Exchange
Health	diabetes, heart disease, disease, heart, illness
Green Energy	Hybrid cars, energy, power, model, carbonated, fuel, bulbs,
Hybrid cars	Hybrid cars, vehicles, model, engines, diesel
Travel	travel, wine, opening, tickets, hotel, sites, cars, search, restaurant
...	...

- 50GB of uncompressed log files
- 50-100M clicks
- 4-6M unique users
- 7000 unique pages with more than 100 hits

Network Monitoring



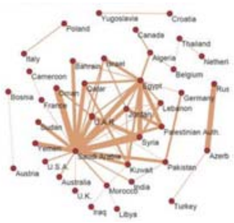
57

Data Statistics

- Leskovec (WWW 2007)
- Log data 150GB/day (compressed)
- 4.5TB of one month data
- Activity over June 2006 (30 days)
  - 245 million users logged in
  - 180 million users engaged in conversation
  - 17 million new account activated
  - More than 30 billion conversation
  - More than 255 billion exchanged messages

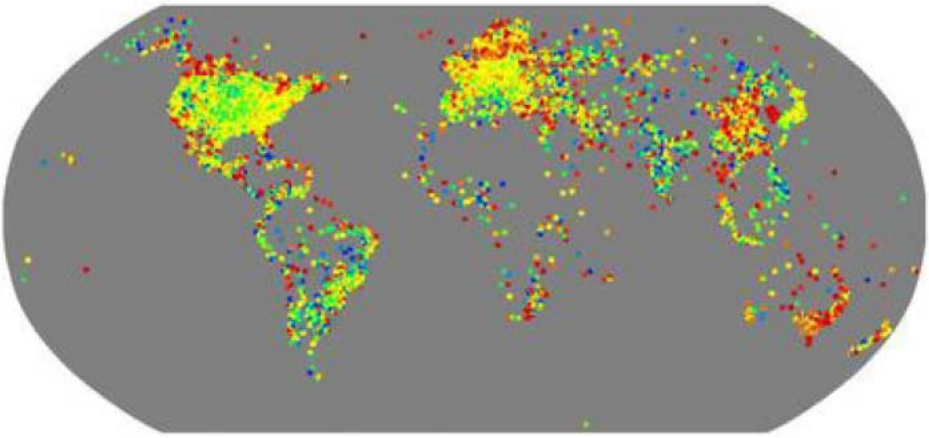
Who talks to whom

who talks to whom (duration)



58

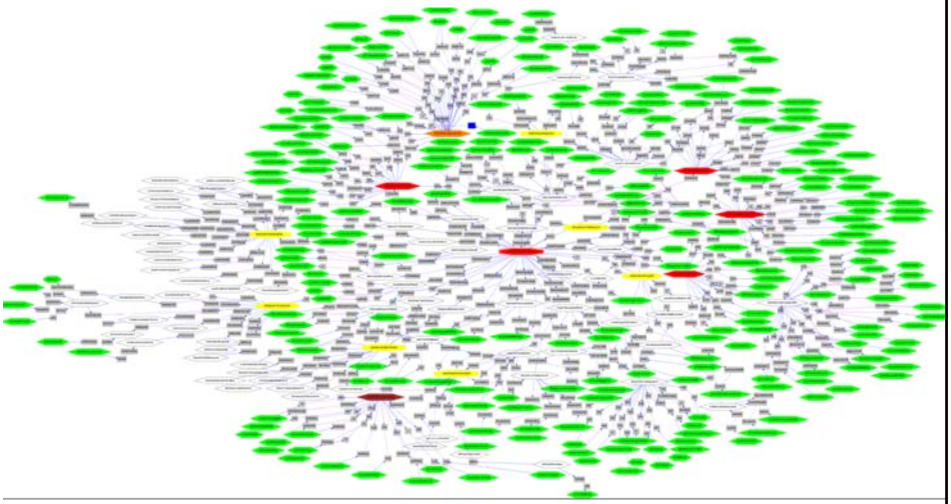
Geography and Communication



Visualisation: News Feed

- http://newsfeed.ijs.si/visual_demo/
- Animation/interactivity often necessary

Visualisation: GraphViz



61

Outline

- What and Why large data?
- Technologies
- Analytics
- Applications
- **Privacy**

62

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Privacy

- Technology is neither good nor bad, it is neutral

- Big data is often generated by people
- Obtaining consent is often impossible
- Anonymisation is very hard…

63

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

You only need 33 bits

- Birth date, postcode, gender
  - Unique for 87% of US population (Sweeney 1997)
- Preference in movies
  - 99% of 500K with 8 rating (Narayanan 2007)
- Web browser
  - 94% of 500K users (Eckersley)
- Writing style
  - 20% accurate out of 100K users (Narayanan 2012)

- How to prevent → Differential Privacy

64

Take Away Messages

- Big Data seems buzz word but it is everywhere
  - Increasing capability of hardware and software will make big data accessible
  - Potential great data analytics
- Can we do big data processing?
  - Yes, but more efficient processing will be required...
- Inter-disciplinary approach is necessary
  - Distributed systems
  - Networking
  - Database
  - Algorithms
  - Machine Learning
- Privacy ! → PART II

65

Acknowledgement

- Joe Bonneau (University of Cambridge)
- Marko Grobelnik Jozef (Stefan Institute)

Thank You!

66

PART II: Big Data and Privacy

Kavé Salamatian
Universite de Savoie

Big Data and privacy

- Data in relational database
  - Linkage attack with auxiliary information
    - e.g. (gender, zip, birthday)
- Matrix data de-anonymization
  - Netflix dataset [NS08]
- Graph data de-anonymization
  - social graph de-anonymization [NS09]

AOL Privacy Debacle

- In August 2006, AOL released anonymized search query logs
  - 657K users, 20M queries over 3 months (March-May)
- Opposing goals
  - Analyze data for research purposes, provide better services for users and advertisers
  - Protect privacy of AOL users
    - Government laws and regulations
    - Search queries may reveal income, evaluations, intentions to acquire goods and services, etc.

AOL User 4417749

- AOL query logs have the form

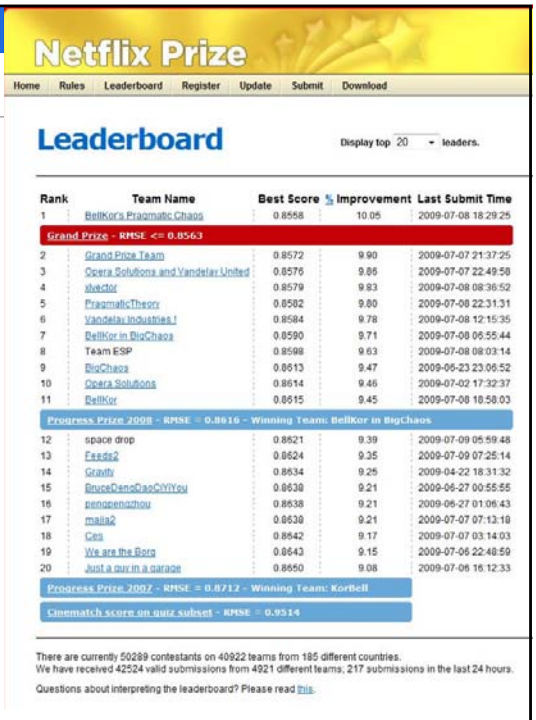
<AnonID, Query, QueryTime, ItemRank, ClickURL>
  - ClickURL is the truncated URL
- NY Times re-identified AnonID 4417749
  - Sample queries: “numb fingers”, “60 single men”, “dog that urinates on everything”, “landscapers in Lilburn, GA”, several people with the last name Arnold
    - Lilburn area has only 14 citizens with the last name Arnold
  - NYT contacts the 14 citizens, finds out AOL User 4417749 is 62-year-old Thelma Arnold

Netflix Prize Dataset

- Netflix: online movie rental service
- In October 2006, released real movie ratings of 500,000 subscribers
  - 10% of all Netflix users as of late 2005
  - Names removed
  - Information may be perturbed
  - Numerical ratings as well as dates
  - Average user rated over 200 movies
- Task is to predict how a user will rate a movie
  - Beat Netflix's algorithm (called Cinematch) by 10%
  - You get 1 million dollars

Netflix Prize

- Dataset properties
  - 17,770 movies
  - 480K people
  - 100M ratings
  - 3M unknowns
- 40,000+ teams
- 185 countries
- $1M for 10% gain



Netflix Prize

Home | Rules | Leaderboard | Register | Update | Submit | Download

Leaderboard

Display top 20 leaders.

Rank	Team Name	Best Score	% Improvement	Last Submit Time
1	BellKor's Pragmatic Chaos	0.8558	10.05	2009-07-08 18:29:25
Grand Prize - RMSE <= 0.8563				
2	Grand Prize Team	0.8572	9.90	2009-07-07 21:37:25
3	Opera Solutions and Vandelay United	0.8576	9.86	2009-07-07 22:49:58
4	xlvector	0.8579	9.83	2009-07-08 08:36:52
5	PragmaticTheory	0.8582	9.80	2009-07-08 22:31:31
6	Vandelay Industries !	0.8584	9.78	2009-07-08 12:15:35
7	BellKor in BigChaos	0.8590	9.71	2009-07-08 06:55:44
8	Team ESP	0.8598	9.63	2009-07-08 08:03:14
9	BigChaos	0.8613	9.47	2009-06-23 23:06:52
10	Opera Solutions	0.8614	9.46	2009-07-02 17:32:37
11	BellKor	0.8615	9.45	2009-07-08 18:58:03
Progress Prize 2008 - RMSE = 0.8616 - Winning Team: BellKor in BigChaos				
12	space drop	0.8621	9.39	2009-07-09 05:59:48
13	Feeds2	0.8624	9.35	2009-07-09 07:25:14
14	Gravity	0.8634	9.25	2009-04-22 18:31:32
15	BruceDengDaoCiYiYou	0.8638	9.21	2009-06-27 00:55:55
16	pengpengzhou	0.8638	9.21	2009-06-27 01:06:43
17	maiia2	0.8638	9.21	2009-07-07 07:13:18
18	Ces	0.8642	9.17	2009-07-07 03:14:03
19	We are the Borg	0.8643	9.15	2009-07-06 22:48:59
20	Just a guy in a garage	0.8650	9.08	2009-07-06 16:12:33
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell				
Cinematch score on quiz subset - RMSE = 0.9514				

There are currently 50289 contestants on 40922 teams from 185 different countries.
We have received 42524 valid submissions from 4921 different teams; 217 submissions in the last 24 hours.

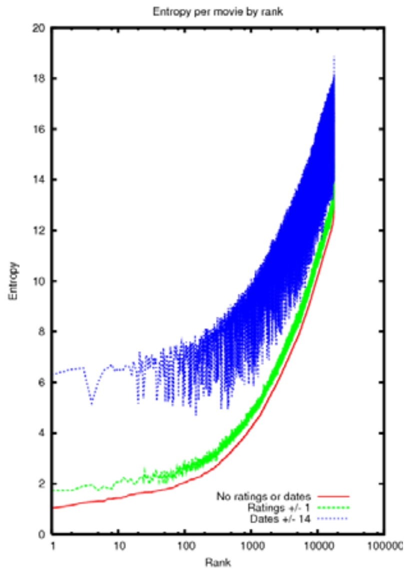
Questions about interpreting the leaderboard? Please read this.

How do you rate a movie?

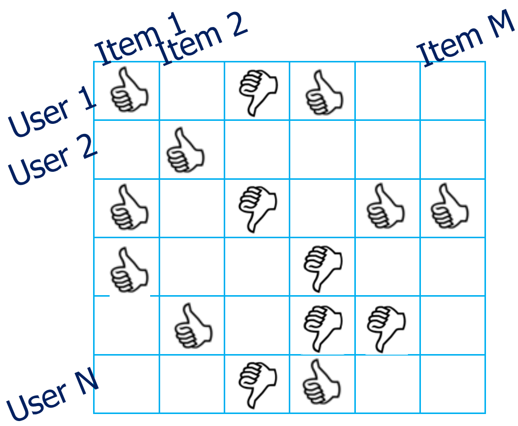
- Report global average
  - I predict you will rate this movie 3.6 (1-5 scale)
  - Algorithm is 15% worse than Cinematch
- Report movie average (Movie effects)
  - Dark knight: 4.3, Wall-E: 4.2, The Love Guru: 2.8, I heart Huckabees: 3.2, Napoleon Dynamite: 3.4
  - Algorithm is 10% worse than Cinematch
- User effects
  - Find each user's average
  - Subtract average from each rating
  - Corrects for curmudgeons and Pollyannas
- Movie + User effects is 5% worse than Cinematch
- More sophisticated techniques use covariance matrix

Netflix Dataset: Attributes

- Most popular movie rated by almost half the users!
- Least popular: 4 users
- Most users rank movies outside top 100/500/1000



Why is Netflix database private?



Provides some anonymity

Privacy question: what can the adversary learn by combining with background knowledge?

No explicit identifiers

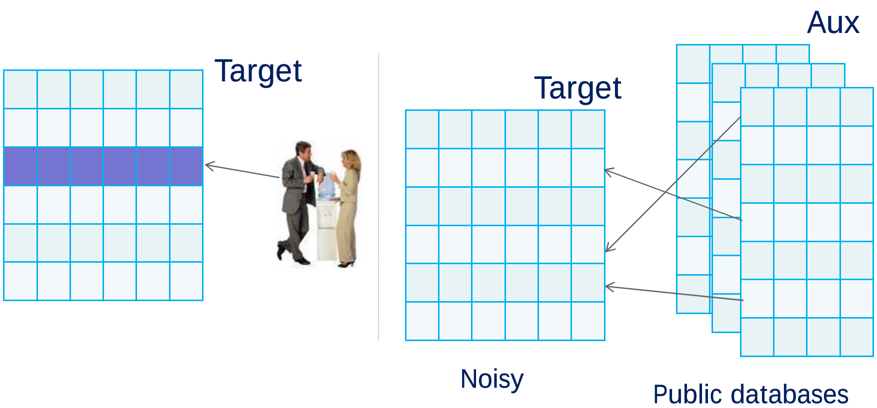
Netflix's Take on Privacy

Even if, for example, you knew all your own ratings and their dates you probably couldn't identify them reliably in the data because only a small sample was included (less than one-tenth of our complete dataset) and that data was subject to perturbation. Of course, since you know all your own ratings that really isn't a privacy problem is it?

-- Netflix Prize FAQ

Background Knowledge (Aux. Info.)

Information available to adversary outside of normal data release process



De-anonymization Objective

- Fix some target record r in the original dataset
- Goal: learn as much about r as possible
- Subtler than “find r in the released database”

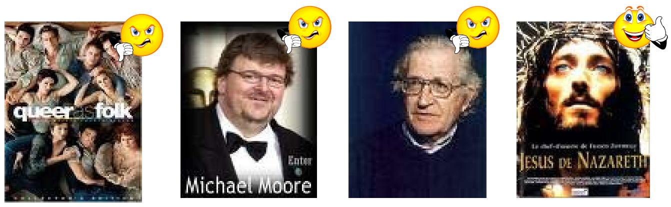
- Background knowledge is noisy
- Released records may be perturbed
- Only a sample of records has been released
- False matches

Narayanan & Shmatikov 2008

martinwilliamrandall	
Email	martinwilliamrandall@yahoo.co.uk
Biography	i went to st peters & st pauls primary from 1982 to 1985

Using IMDb as Aux

- Extremely noisy, some data missing
- Most IMDb users are not in the Netflix dataset
- Here is what we learn from the Netflix record of one IMDb user (<u>not</u> in his IMDb profile)

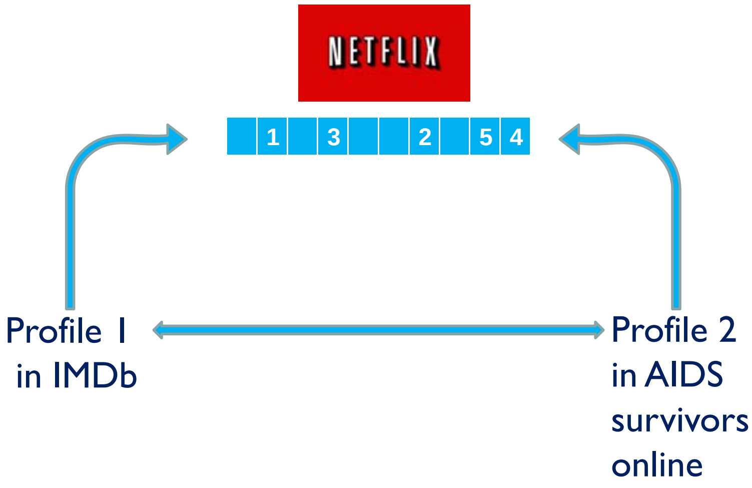


De-anonymizing the Netflix Dataset

- Average subscriber has 214 dated ratings
- Two is enough to reduce to 8 candidate records
- Four is enough to identify uniquely (on average)
- Works even better with relatively rare ratings
  - "The Astro-Zombies" rather than "Star Wars"

Fat Tail effect helps here:
most people watch obscure movies
(really!)

More linking attacks



Anonymity vs. Privacy

Anonymity is insufficient for privacy

Anonymity is necessary for privacy

Anonymity is unachievable in practice

Re-identification attack → anonymity breach → privacy breach

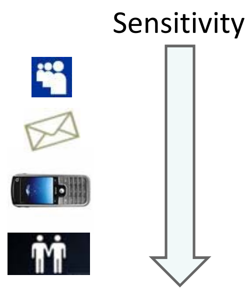
Just ask Justice Scalia
“It is silly to think that every single
datum about my life is private”

Beyond recommendations…

- Adaptive systems reveal information about users

Social Networks

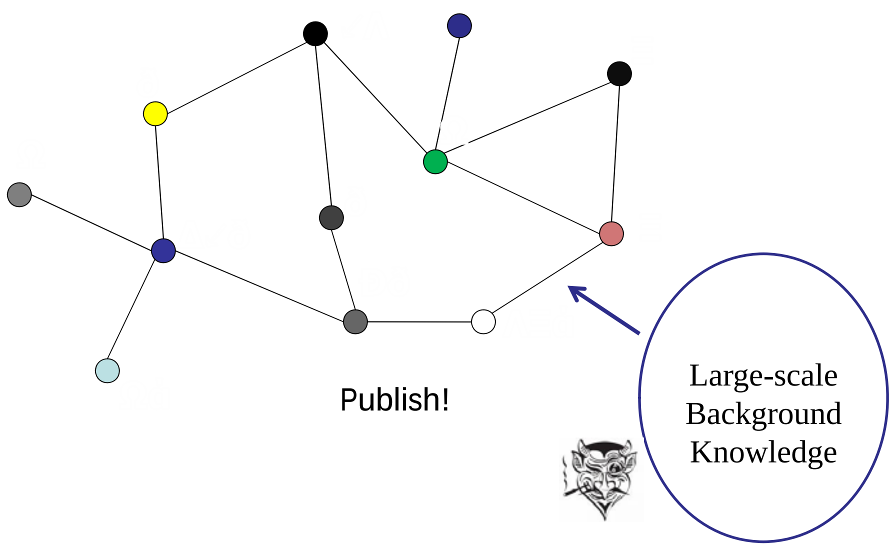
- Online social network services
- Email, instant messenger
- Phone call graphs
- Plain old real-life relationships



Social Networks: Data Release

Select subset of nodes → Compute induced subgraph → Sanitize edges → Select attributes → Publish

Attack Model



Motivating Scenario: Overlapping Networks

- Social networks A and B have overlapping memberships
- Owner of A releases anonymized, sanitized graph
  - say, to enable targeted advertising
- Can owner of B learn sensitive information from released graph A'?

Re-identification: Two-stage Paradigm

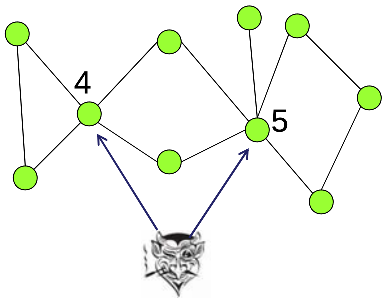
Re-identifying target graph =
Mapping between Aux and target nodes

- Seed identification:
  - Detailed knowledge about small number of nodes
  - Relatively precise
  - Link neighborhood constant
  - In my top 5 call and email list…..my wife
- Propagation: similar to infection model
  - Successively build mappings
  - Use other auxiliary information
    - I'm on facebook and flickr from 8pm-10pm
- Intuition: no two random graphs are the same
  - Assuming enough nodes, of course

Seed Identification: Background Knowledge

How:

- Creating sybil nodes
- Bribing
- Phishing
- Hacked machines
- Stolen cellphones



What: List of neighbors

- Degree
- Number of common neighbors of two nodes

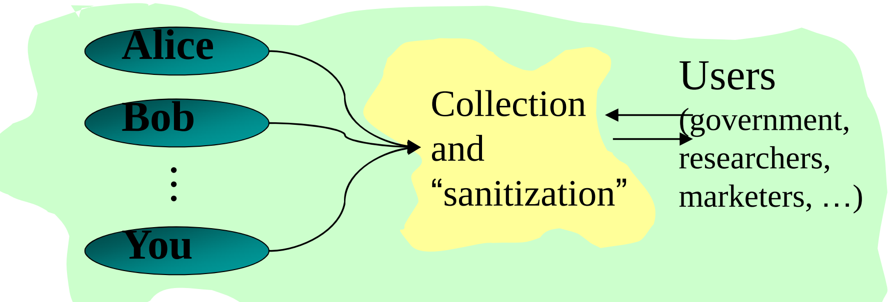
Degrees: (4,5)
Common nbrs: (2)

Preliminary Results

- Datasets: flickr twitter
- 27,000 common nodes
- Only 15% edge overlap
- 150 seeds
- 32% re-identified as measured by centrality
  - 12% error rate

Solutions

Database Privacy

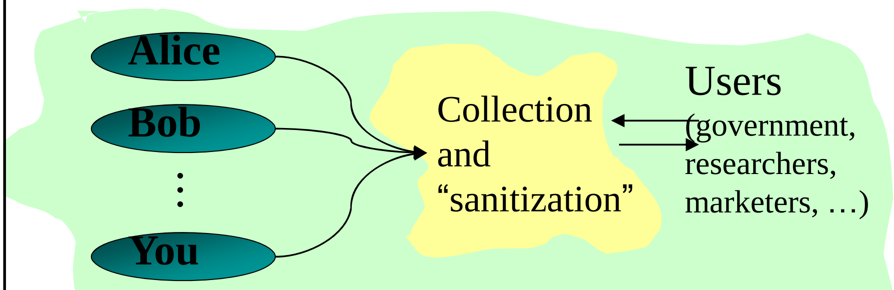


"Census problem"

Two conflicting goals

- **Utility**: Users can extract "global" statistics
- **Privacy**: Individual information stays hidden
- How can these be formalized?

Database Privacy



Variations on model studied in

- Statistics
- Data mining
- Theoretical CS
- Cryptography

Different traditions for what "privacy" means

How can we formalize "privacy"?

- Different people mean different things
- Pin it down mathematically?

Goal #1: Rigor

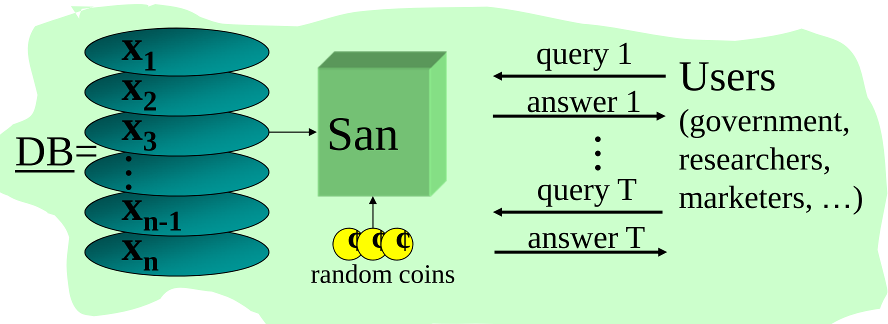
- Prove clear theorems about privacy
  - Few exist in literature
- Make clear (and refutable) conjectures
- Sleep better at night

Goal #2: Interesting science

- (New) Computational phenomenon
- Algorithmic problems
- Statistical problems

96

Basic Setting



- Database DB = table of n rows, each in domain D
  - D can be numbers, categories, tax forms, etc
  - E.g.: Married?, Employed?, Over 18?, …

Examples of sanitization methods

- Input perturbation
  - Change data before processing
  - E.g. Randomized response
    - flip each bit of table with probability *p*
- Summary statistics
  - Means, variances
  - Marginal totals (# people with blue eyes and brown hair)
  - Regression coefficients
- Output perturbation
  - Summary statistics with noise
- Interactive versions of above:
  - **Auditor** decides which queries are OK, type of noise

Two Intuitions for Privacy

"If the release of statistics *S* makes it possible to determine the value [of private information] more accurately than is possible without access to *S*, a disclosure has taken place." [Dalenius]

- Learning **more** about me should be hard

Privacy is "protection from being **brought to the attention of others**." [Gavison]

- Safety is blending into a crowd

- Problems with Classic Intuition
  - Popular interpretation: prior and posterior views about an individual shouldn't change "too much"
    - What if my (incorrect) prior is that every UTCS graduate student has three arms?

- How much is "too much?"
  - Can't achieve cryptographically small levels of disclosure and keep the data useful
  - Adversarial user is supposed to learn unpredictable things about the database

Straw man: Learning the distribution

- Assume $x_1,\ldots,x_n$ are drawn i.i.d. from unknown distribution
- **Def'n**: San is safe if it only reveals **distribution**
- Implied approach:
  - learn the distribution
  - release description of distrib
  - or re-sample points from distrib
- Problem: tautology trap
  - estimate of distrib. depends on data… why is it safe?

Blending into a Crowd

- Intuition: I am safe in a group of **k** or more
  - **k** varies (3… 6… 100… 10,000 ?)
- Many variations on theme:
  - Adv. wants predicate **g** such that

    $0 < \#\{\, i \mid g(x_i)=\text{true} \,\} < k$
  - ***g*** is called a **breach** of privacy
- Why?
  - Fundamental:
    - R. Gavison: "protection from being brought to the attention of others"
  - Rare property helps me **re-identify** someone
  - **Implicit**: information about a **large group** is public
    - e.g. liver problems more prevalent among diabetics

101

Blending into a Crowd

- Intuition: I am safe in a group of **k** or more
  - **k** varies (3… 6… 100… 10,000 ?)
- Many variations on theme:
  - Adv. wants predicate **g** such that

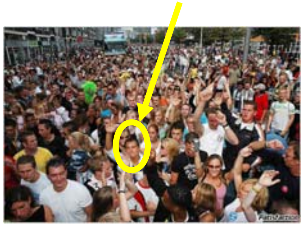
    $0 < \#\{\, i \mid g(x_i)=\text{true} \,\} < k$
  - ***g*** is called a **breach** of privacy
- Why?
  - Fundamental:
    - R. Gavison: "
  - Rare property hel
  - **Implicit**: informati
    - e.g. liver prob

Two variants:
- frequency in DB
- frequency in underlying population

How can we capture this?
- Syntactic definitions
- Bayesian adversary
- "Crypto-flavored" definitions

Blending into a Crowd

- Intuition: I am safe in a group of **k** or more
- pros:
  - appealing intuition for privacy
  - seems fundamental
  - mathematically interesting
  - meaningful statements are possible!



- cons
  - does it rule out learning facts about particular individual?
  - **all results** seem to make strong assumptions on adversary's prior distribution
  - is this **necessary?** (yes…)

Impossibility Result

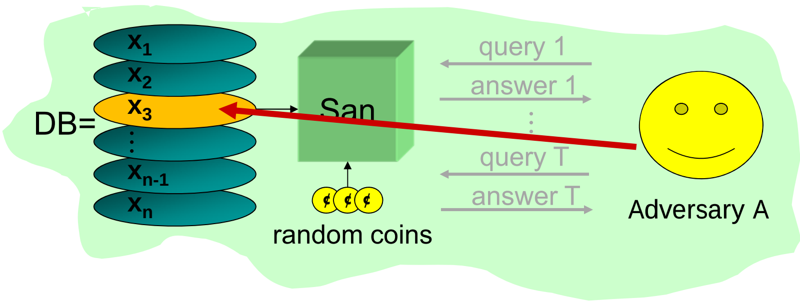
[Dwork]

- Privacy: for some definition of "privacy breach,"

  ∀ distribution on databases, ∀ adversaries A, ∃ A'

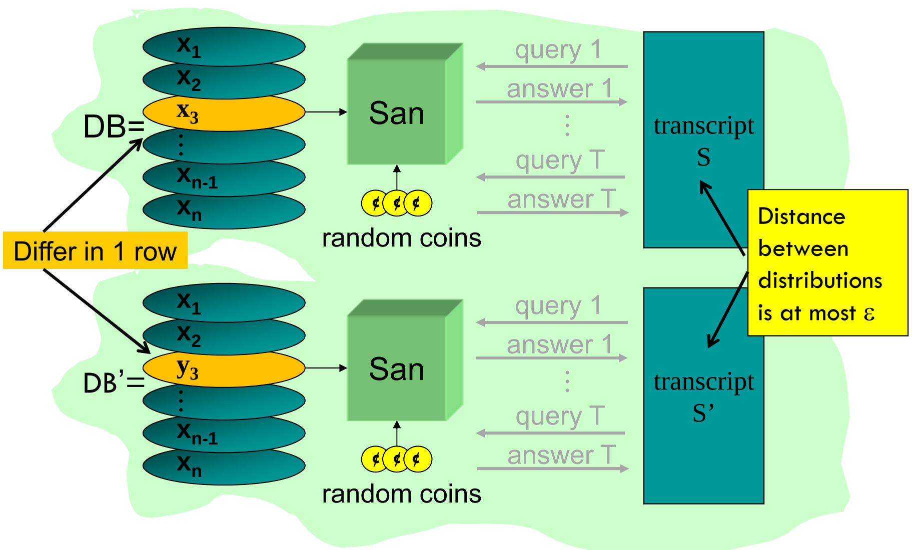
  such that $\Pr(A(San)=breach) - \Pr(A'()=breach) \leq \varepsilon$
  - For reasonable "breach", if San(DB) contains information about DB, then some adversary breaks this definition
- Example
  - Vitaly knows that Josh Leners is 2 inches taller than the average Russian
  - DB allows computing average height of a Russian
  - This DB breaks Josh's privacy according to this definition… even if his record is not in the database!

Differential Privacy (1)

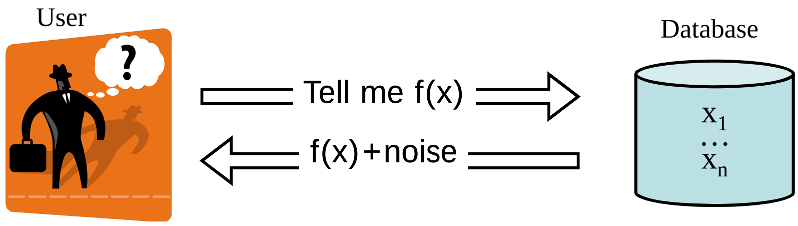


- Example with Russians and Josh Leners
  - Adversary learns Josh's height even if he is not in the database
- Intuition: "Whatever is learned would be learned regardless of whether or not Josh participates"
  - Dual: Whatever is already known, situation won't get worse

Indistinguishability



Diff. Privacy in Output Perturbation

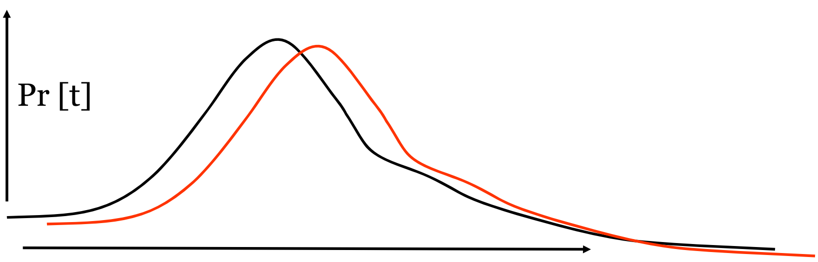


- Intuition: f(x) can be released accurately when f is insensitive to individual entries $x_1, \ldots x_n$
- Global sensitivity $GS_f = \max_{\text{neighbors } x,x'} ||f(x) - f(x')||_1$ Lipschitz constant of f
  - Example: $GS_{average} = 1/n$ for sets of bits
- Theorem: f(x) + Lap(GS_f / ε) is ε-indistinguishable
  - Noise generated from Laplace distribution

Differential Privacy: Summary

- K gives ε-differential privacy if for all values of DB and Me and all transcripts t:

$$\frac{\Pr[\mathcal{K}(DB - Me) = t]}{\Pr[\mathcal{K}(DB + Me) = t]} \leq e^{\varepsilon} \approx 1 \pm \varepsilon$$



Why does this help?

With relatively little noise:

- Averages
- Histograms
- Matrix decompositions
- Certain types of clustering
- …

Preventing Attribute Disclosure

- Various ways to capture
  "no particular value should be revealed"
- Differential Criterion:
  - "Whatever is learned would be learned regardless of whether or not person **i** participates"
- Satisfied by indistinguishability
  - Also implies protection from re-identification?
- Two interpretations:
  - A given release won't make privacy worse
  - Rational respondent will answer if there is some gain
- Can we preserve enough utility?

Thanks to

Thanks to Vitaly Shmatikov, James Hamilton
Salman Salamatian