CPT-S 415

Big Data

Yinghui Wu

EME B45

CPT-S 415
Big Data

Data type, challenges and research overview

- ✓ Data types and data representation
- ✓ Big data challenge/fallacies: from the eye of computation
- ✓ Research topics in Big Data Management
- ✓ Project overview

Data and data types

What is Data?

- Collection of data objects and their attributes
- An attribute is a property or characteristic of an object
  - Examples: eye color of a person, temperature, etc.
  - Attribute is also known as variable, field, characteristic, or feature
- A collection of attributes/values describe an object
  - Object is also known as record, point, case, sample, entity, or instance

Attributes

Objects

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Attribute Values

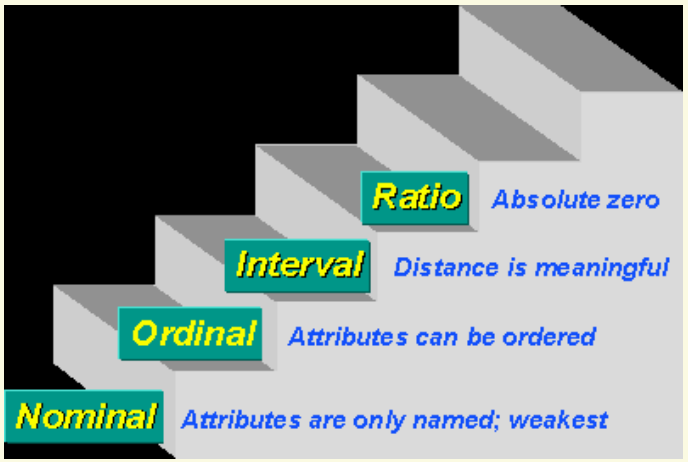
- ✓ Attribute values are numbers or symbols assigned to an attribute
- ✓ Attributes vs. attribute values
  - Same attribute can be mapped to different attribute values
    - Example: **height** can be measured in feet or meters
  - Different attributes can be mapped to the same set of values
    - Example: Attribute values for ID and age are integers
    - properties of attribute values can be different
      - ID has no limit but age has a maximum and minimum value
      - Partly described as domain (domain constraints) of the attribute

Level of attribute measures

- Types of attributes (data measurement)
  - Nominal (categorical): attributes are only named, weakest
    - Examples: weather (sunny, rainy), eye color, zip codes
  - Ordinal: attributes can be ordered
    - Examples: rankings (e.g., taste of potato chips on a scale from 1-10), grades, height in {tall, medium, short}
  - Interval: distance is meaningful
    - Examples: calendar dates, temperatures in Celsius or Fahrenheit.
  - Ratio: absolute zero – meaningful fraction/ratio
    - Examples: temperature in Kelvin, length, time, counts

Properties of attribute

- The type of an attribute depends on its properties:
  - Distinctness: = ≠
  - Order: < >
  - Addition: + -
  - Multiplication: * /



  - Nominal attribute: distinctness
  - Ordinal attribute: distinctness & order
  - Interval attribute: distinctness, order & addition
  - Ratio attribute: all 4 properties

Attribute types

Attribute Type	Description	Examples	Operations
Nominal	The values of a nominal attribute are just different names, i.e., nominal attributes provide only enough information to distinguish one object from another. (=, ≠)	zip codes, employee ID numbers, eye color, sex: {*male, female*}	mode, entropy, contingency correlation, χ^2 test
Ordinal	The values of an ordinal attribute provide enough information to order objects. (<, >)	hardness of minerals, {*good, better, best*}, grades, street numbers	median, percentiles, rank correlation, run tests, sign tests
Interval	differences between values are meaningful, i.e., a unit of measurement exists. (+, -)	calendar dates, temperature in Celsius or Fahrenheit	mean, standard deviation, Pearson's correlation, *t* and *F* tests
Ratio	For ratio variables, both differences and ratios are meaningful. (*, /)	temperature in Kelvin, monetary quantities, counts, age, mass, length, electrical current	geometric mean, harmonic mean, percent variation

Discrete and Continuous Attributes

- Discrete Attribute
  - Has only a finite or countably infinite set of values
  - Examples: zip codes, counts, or the set of words in a collection of documents
  - Often represented as integer variables.
  - Note: binary attributes are a special case of discrete attributes

- Continuous Attribute
  - Has real numbers as attribute values
  - Examples: temperature, height, or weight.
  - Practically, real values can only be measured and represented using a finite number of digits.
  - Continuous attributes are typically represented as floating-point variables.

Type of attributes: database perspective

- ✓ Simple vs composite
  - Salary, age,
  - Address (house no: city: state), Name (first name: last name)
- ✓ Single valued vs. multi valued
  - a person can only have one ‘date of birth’, ‘age’
  - A customer can have multiple phone numbers
- ✓ Stored vs derived
  - a stored attribute provides value to related attribute. DOB
  - a derived attribute: age = current date – DOB
- ✓ Complex attribute: both composite and multi valued.

Types of data sets

- Record
  - Data Matrix
  - Document Data
  - Transaction Data
- Ordered
  - Spatial Data
  - Temporal Data
  - Sequential Data
  - Genetic Sequence Data
- Graph/Network/Linked
  - World Wide Web
  - Molecular Structures

Record Data

- Data that consists of a collection of records, each of which consists of a fixed set of attributes

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Record: Data Matrix

- Data objects as points in a multi-dimensional space, where each dimension represents a distinct attribute
  - Such data set can be represented by an m by n matrix, where there are m rows, one for each object, and n columns, one for each attribute

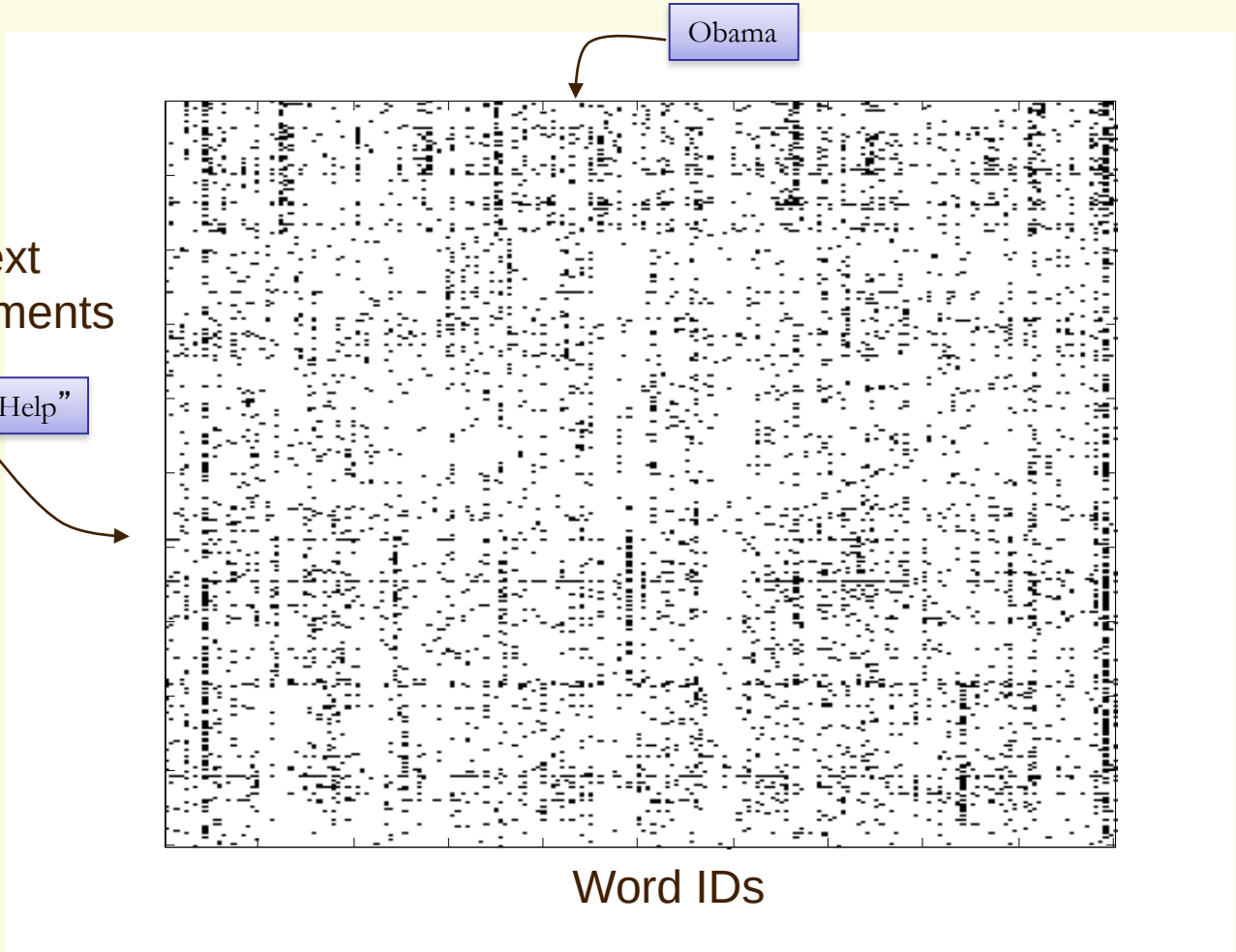
Projection of x Load	Projection of y load	Distance	Load	Thickness
10.23	5.27	15.22	2.7	1.2
12.65	6.25	16.22	2.2	1.1

Record: Document Data

- Each document becomes a `term' vector,
  - each term is a component (attribute) of the vector,
  - the value of each component is the number of times the corresponding term occurs in the document.

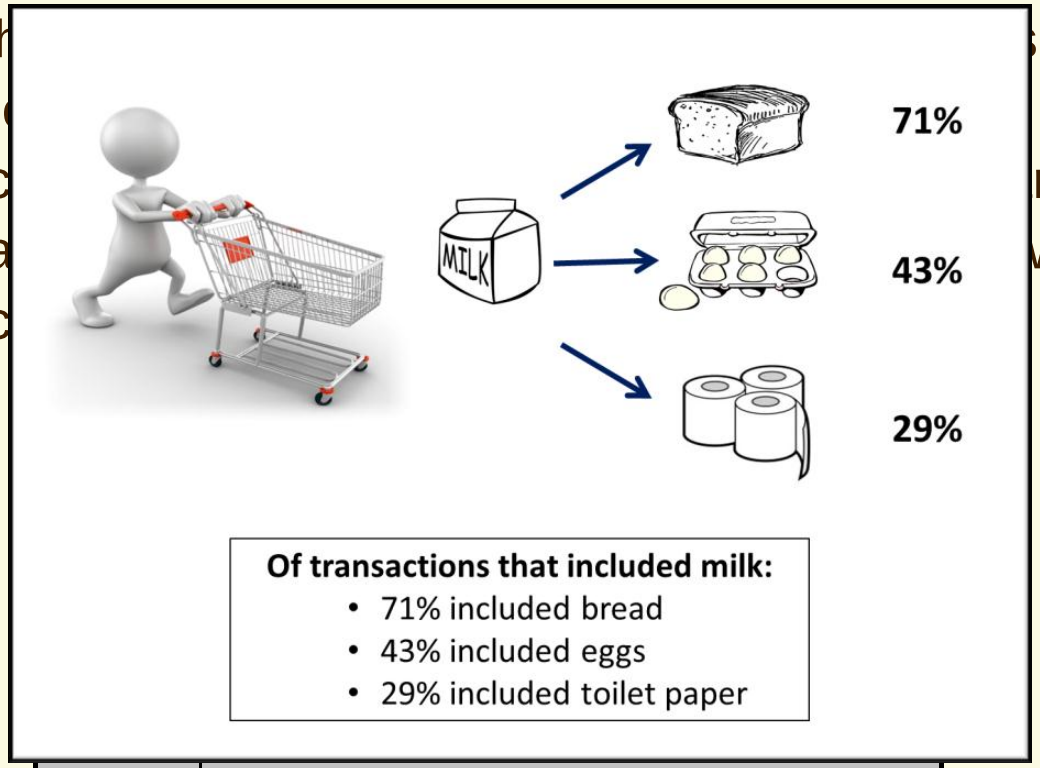
	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

A sparse matrix of Text Data



Record: Transaction Data

- A special type of record data, where
  - each
  - For
    purc
    a tra
    purc



5	Coke, Diaper, Milk

Record: Transaction Data

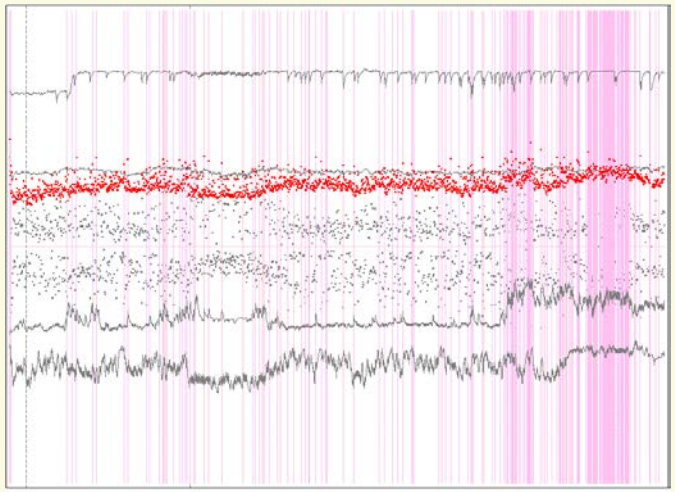
Date stamped events (weblogs, phone calls):

128.195.36.195, -, 3/22/00, 10:35:11, W3SVC, SRVR1, 128.200.39.181, 781, 363, 875, 200, 0, GET, /top.html, -,
128.195.36.195, -, 3/22/00, 10:35:16, W3SVC, SRVR1, 128.200.39.181, 5288, 524, 414, 200, 0, POST, /spt/main.html, -,
128.195.36.195, -, 3/22/00, 10:35:17, W3SVC, SRVR1, 128.200.39.181, 30, 280, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
128.195.36.101, -, 3/22/00, 16:18:50, W3SVC, SRVR1, 128.200.39.181, 60, 425, 72, 304, 0, GET, /top.html, -,
128.195.36.101, -, 3/22/00, 16:18:58, W3SVC, SRVR1, 128.200.39.181, 8322, 527, 414, 200, 0, POST, /spt/main.html, -,
128.195.36.101, -, 3/22/00, 16:18:59, W3SVC, SRVR1, 128.200.39.181, 0, 280, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
128.200.39.17, -, 3/22/00, 20:54:37, W3SVC, SRVR1, 128.200.39.181, 140, 199, 875, 200, 0, GET, /top.html, -,
128.200.39.17, -, 3/22/00, 20:54:55, W3SVC, SRVR1, 128.200.39.181, 17766, 365, 414, 200, 0, POST, /spt/main.html, -,
128.200.39.17, -, 3/22/00, 20:54:55, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
128.200.39.17, -, 3/22/00, 20:55:07, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
128.200.39.17, -, 3/22/00, 20:55:36, W3SVC, SRVR1, 128.200.39.181, 1061, 382, 414, 200, 0, POST, /spt/main.html, -,
128.200.39.17, -, 3/22/00, 20:55:36, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
128.200.39.17, -, 3/22/00, 20:55:39, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
128.200.39.17, -, 3/22/00, 20:56:03, W3SVC, SRVR1, 128.200.39.181, 1081, 382, 414, 200, 0, POST, /spt/main.html, -,
128.200.39.17, -, 3/22/00, 20:56:04, W3SVC, SRVR1, 128.200.39.181, 0, 258, 111, 404, 3, GET, /spt/images/bk1.jpg, -,
128.200.39.17, -, 3/22/00, 20:56:33, W3SVC, SRVR1, 128.200.39.181, 0, 262, 72, 304, 0, GET, /top.html, -,
128.200.39.17, -, 3/22/00, 20:56:52, W3SVC, SRVR1, 128.200.39.181, 19598, 382, 414, 200, 0, POST, /spt/main.html, -,

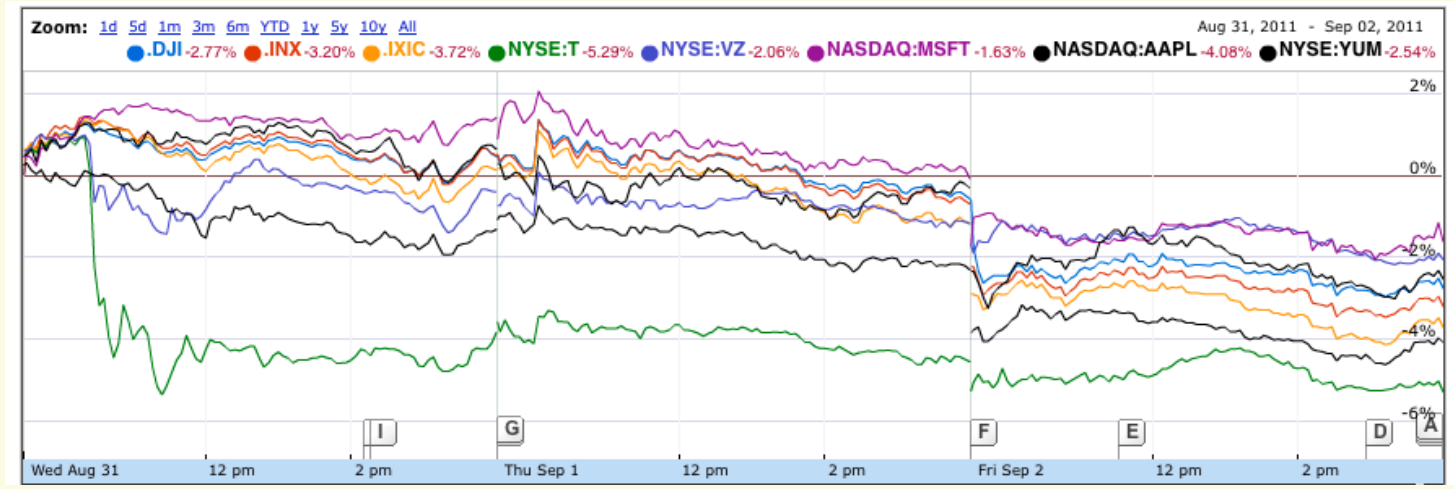
Can be represented as a time series:

User 1	2	3	2	2	3	3	3	1	1	1	3	1	3	3	3	3
User 2	3	3	3	1	1	1										
User 3	7	7	7	7	7	7	7	7								
User 4	1	5	1	1	1	5	1	5	1	1	1	1	1	1		
User 5	5	1	1	5												
…		…														

Ordered Data: Sequential Data



Often many time series, long time series, or multivariate time series



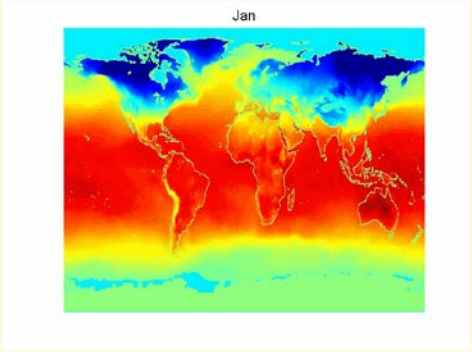
Ordered Data: Genomic sequence data

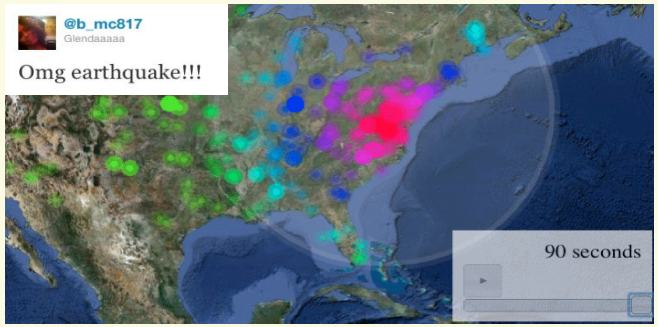
http://www.ncbi.nlm.nih.gov/genbank/

```
>PB22887-RA class=Sequence position=scf7180000350335:1110961..1112267 (+ strand)
ATTCTTTCTT TCAGCGTTCA TCCTACGAGA AGGTATGTCA TATCAGATTA TATTTCGAAA TATGTATATC GTGTTTTATA
TATAGTTTTC TAAGTATGTA GAGGATTAAT AAATTATAAA TGTTATTACA GTCGAGGAAG CATAAAACCA CAAAAATGGT
AGCTCAAAAG AAACAGGTAA TTACACTGTT GCACGTCCAC GTGTTCGTCC GTAAGTTTTC TCGTAGCAAA ATTTCATGCT
TTTCAATATA TGCGTAGATA TGATCTTGTT ATATCATTAT TATAATTGTT TTAACTGTAA ATCGTTGCTT TTACATTGAT
TGAAATAATA TAAAACTCGA GTGTACCATA GTAATCTGAA AGGAACCGCT TGATTTTGCT ATTCTCTTTC TAAATTTTGC
TTCTATCTAC TATGAGAAAT ATAGATTCTA TTTTCAATTT CTCAAATCTA CAAATCGTAT CGCTATTCTG ATCGACAATT
TATAATTTCT TGTTACTTCA GAACGTGAGT TATTAAGTTA ATCTAAATAT ATCAATGTAT GTAATGAGTT TTATTACCTG
ATCATATAGA AAAAGGCTCA GGAGAGCATC AACACCAGAT TAGCACTCGT CATGAAATCT GGTAAATACG TCCTTGGTTA
TAAACAAACT CTGAAGTCAC TCCGCCAAGG CAAAGCTAAA TTGGTCATCA TTGCTAGCAA TACGCCACCG CTAAGGTGAG
ACTGAGAAGT AGCACTCCTT TTGTATTGGC AAATTGACAT ATTTAAAATA AATAATTTCT TTTATAAGCG AATTAATTCA
AAATTTAATG AATATAATCT TTGATAAATG TATACATATA TATATACATA TATTTGTAAA AATTATATTC TATTGTAATT
TAATTTTATT AGAATCTAGT GAAATATTAA ATAAATAATT ACATTAGAGA TGCAGTAATA GAAACTAGTA ATATTGATAT
AATCATCAAT TTGCATATTG GAAAACAAAT AGGAGTACTA TGGAGTAGTC TCCAGCTCCT GCTTGTAAGT GCATGGGGCA
TTAAGATTTA ATTTTTTAAT GTTTTAAACA TAATTTATCT CGATTTAACA GGAAGTCGGA GATTGAATAC TATGCAATGT
TAGCGAAGAC TGGTGTGCAT CATTACACGG GGAATAACAT CGAACTGGGT ACAGCTTGTG GTAAATATTT CCGTGTCTGT
ACACTCTCGA TCACAGATCC TGGTAATTCT GACATTATAA AATCTATGCC AACTGGTGAT CAAGCGTAAT GTACAGTTTT
TAATCCAATA AATAATTCAA AACGTTT
```

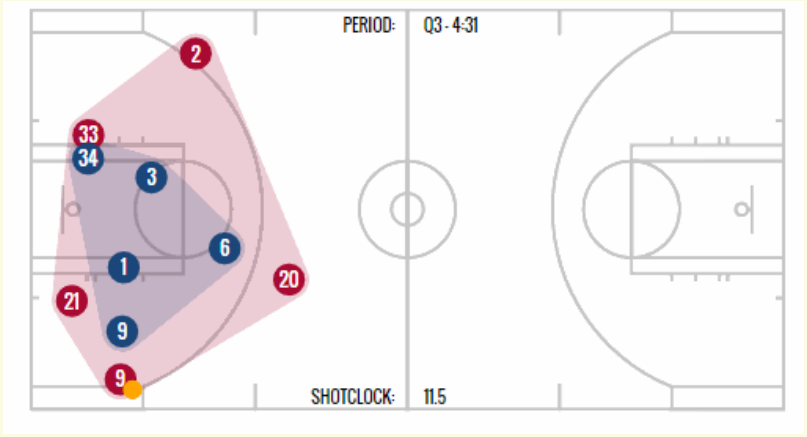
Ordered Data: Spatio-Temporal Data

- ✓ Average Monthly Temperature of land and ocean
- ✓ Earthquake data



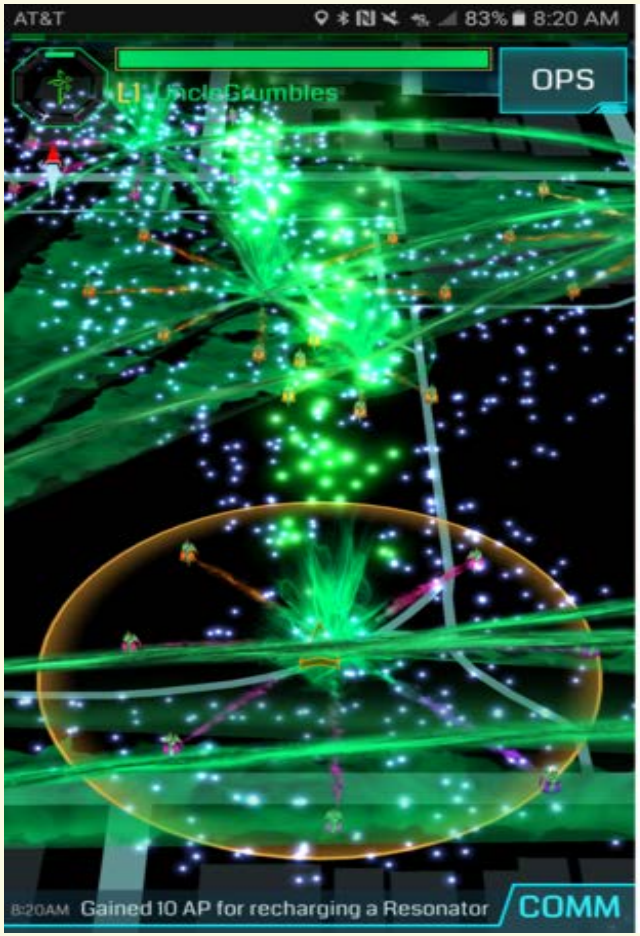


- ✓ Sports data



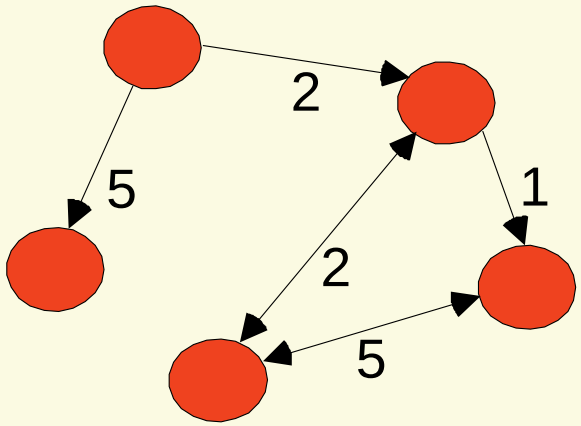
http://stats.nba.com/featured/movement_animations_2014_10_27.html

Ordered Data: Spatio-Temporal Data



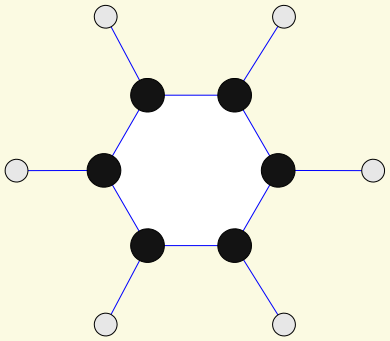
Graph & Network Data

- Generic graph and HTML Links

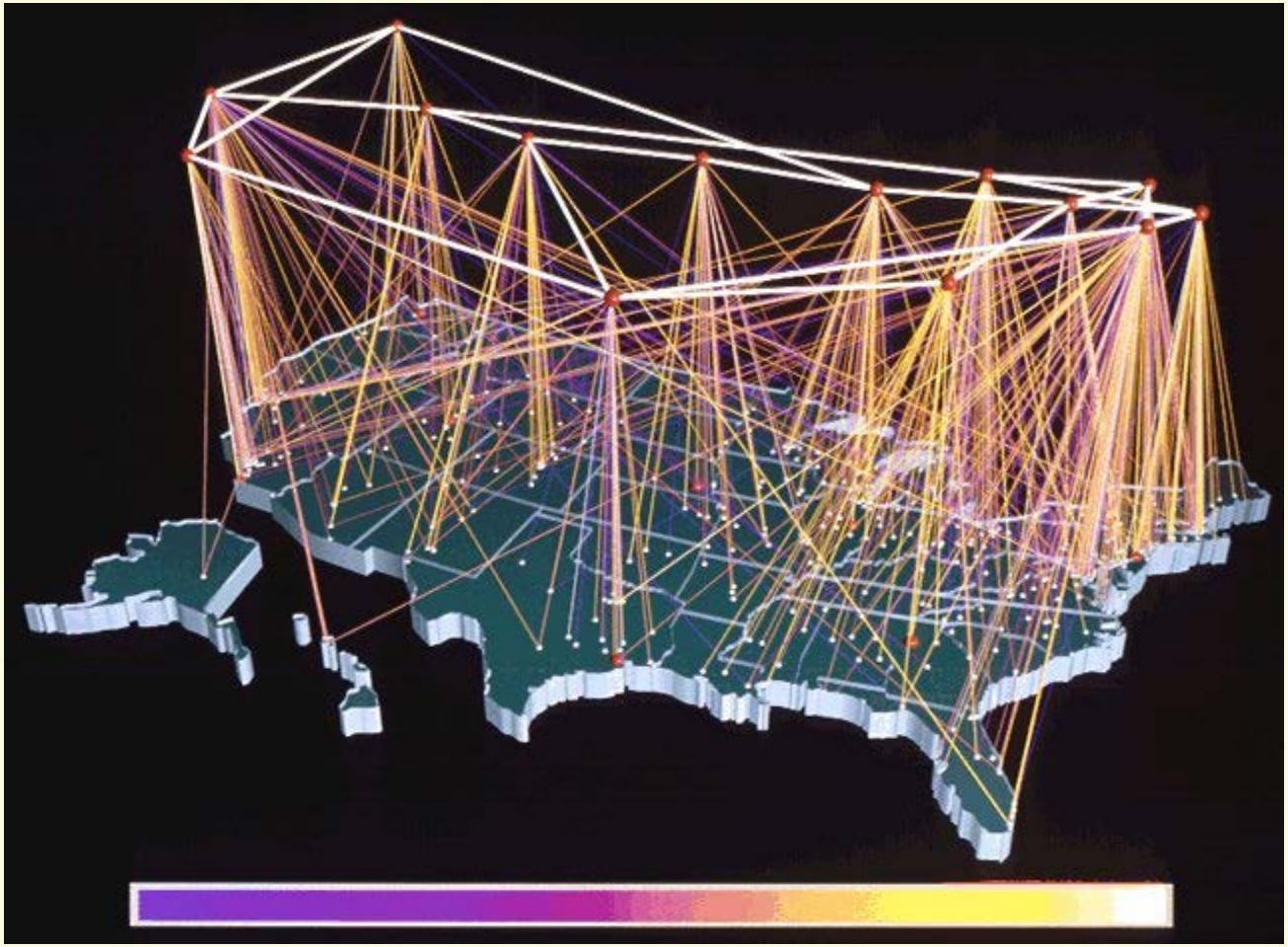



```
<a href="papers/papers.html#bbbb">
Data Mining </a>
<li>
<a href="papers/papers.html#aaaa">
Graph Partitioning </a>
<li>
<a href="papers/papers.html#aaaa">
Parallel Solution of Sparse Linear System of Equations </a>
<li>
<a href="papers/papers.html#ffff">
N-Body Computation and Dense Linear System Solvers
```

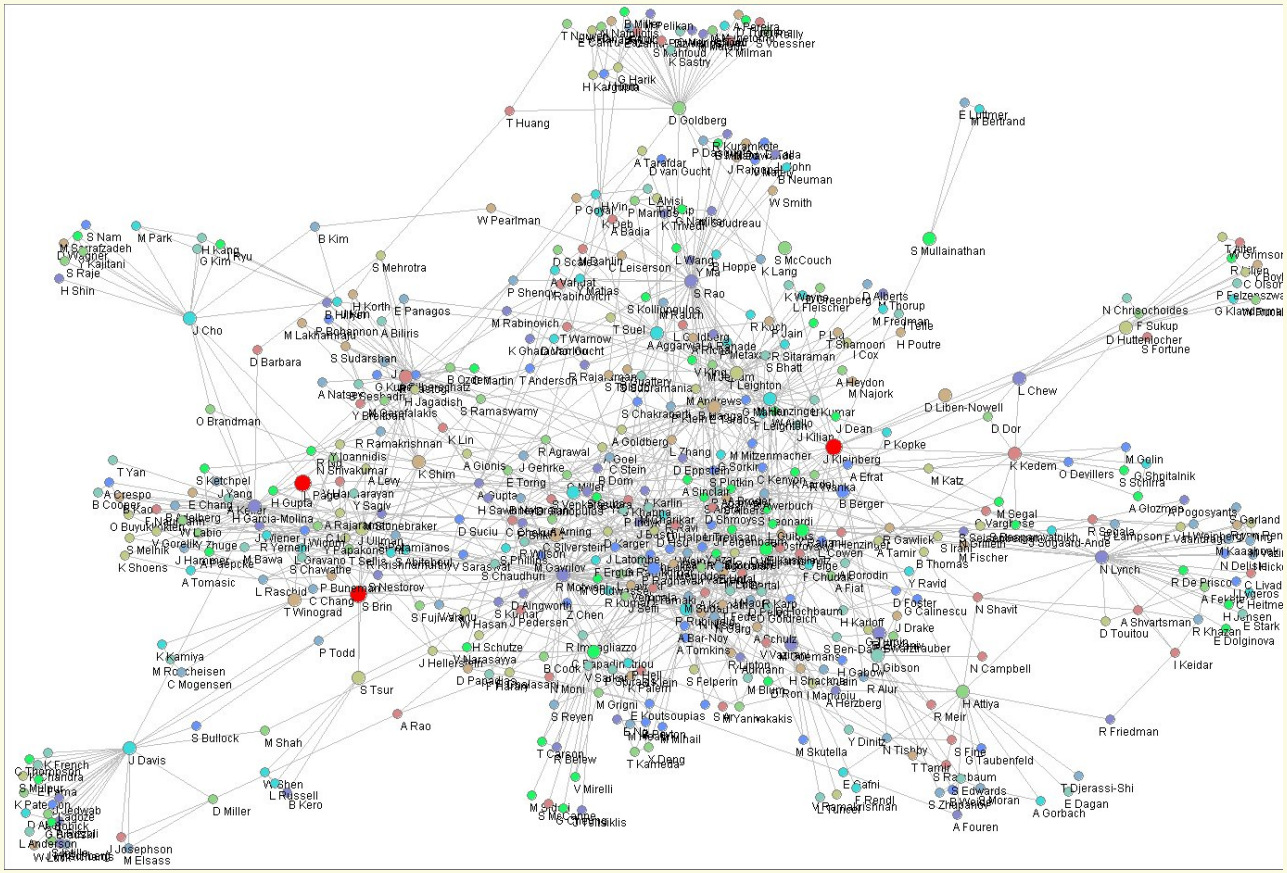
Benzene Molecule: C_6H_6



Network Data: Physical Network

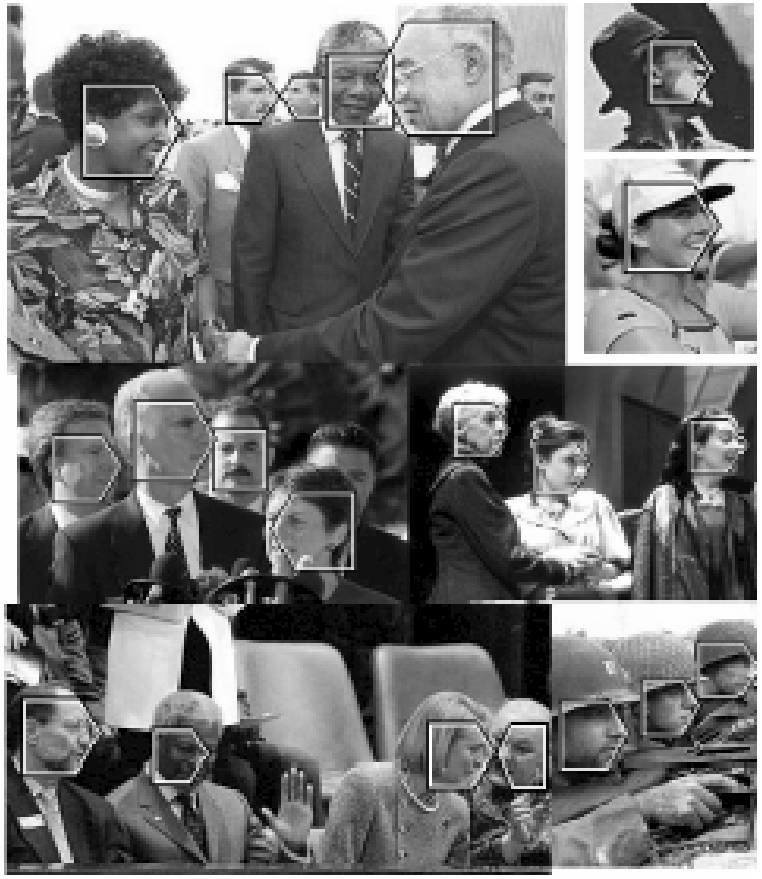


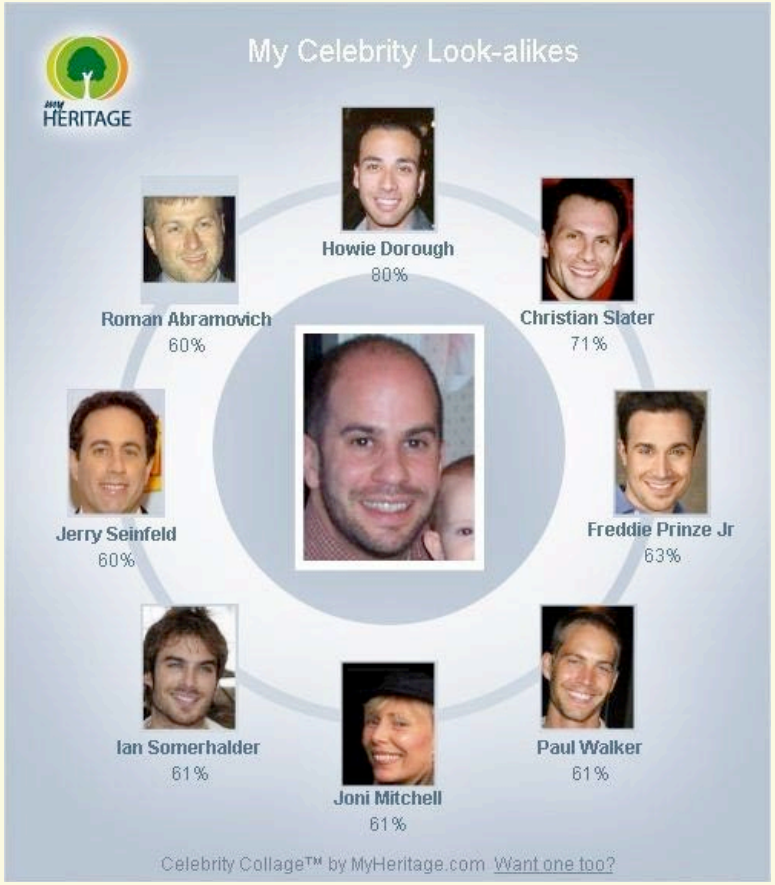
Network Data: Derived Social Network



Algorithms for estimating relative importance in networks
S. White and P. Smyth, *ACM SIGKDD*, 2003.

Image Data




My Celebrity Look-alikes
HERITAGE
Howie Dorough
80%
Roman Abramovich
60%
Christian Slater
71%
Jerry Seinfeld
60%
Freddie Prinze Jr
63%
Ian Somerhalder
61%
Joni Mitchell
61%
Paul Walker
61%
Celebrity Collage™ by MyHeritage.com Want one too?

big data challenge: from the eye of computation

Big data: Through the eyes of computation

- ✓ Computer science is the topic about

the computation of function f(x)

- ✓ Big data: the data parameter *x* is horrendously large: PB or EB

What is the challenge introduced to query answering?

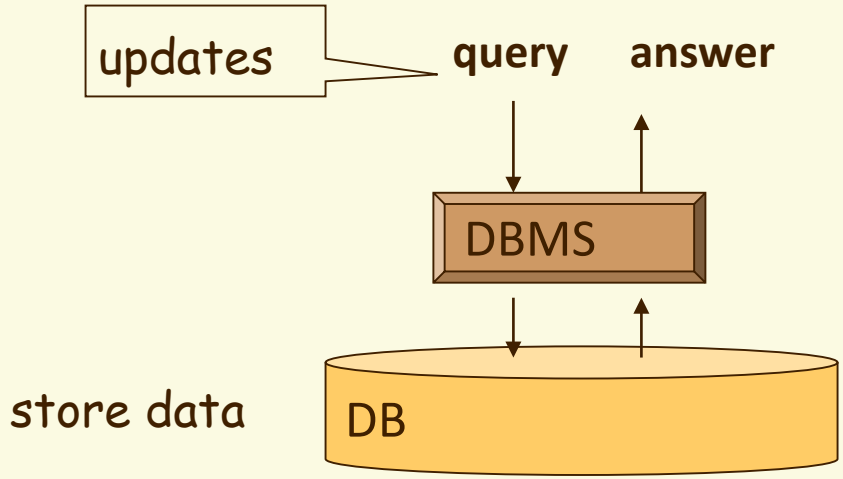
Fallacies:

- ✓ *Big data introduces no fundamental problems*
- ✓ *Big data = MapReduce (Hadoop)*
- ✓ *Big data = data quantity (scalability)*

Are these true?

Traditional database management systems

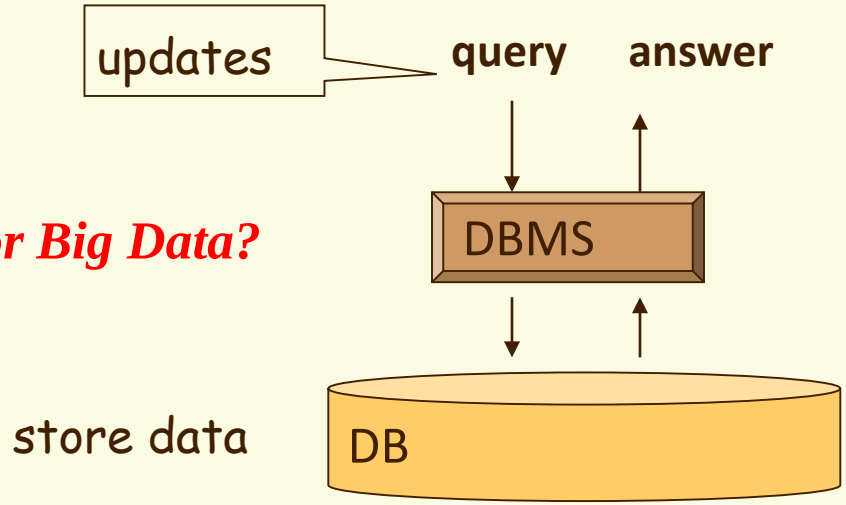
- A ***database*** is a collection of data, typically containing the information about one or more related organizations.
- A ***database management system*** (***DBMS***) is a software package designed to store and manage databases.
- Database: local
- DBMS: centralized; single processor (CPU); managing local databases (single memory, disk)



Relational queries

Keywords in traditional database research:

- ✓ What is a relational schema? A relation? A relational database?
- ✓ What is a query? What is relational algebra?
- ✓ What does relationally completeness mean?
- ✓ What is a conjunctive query?

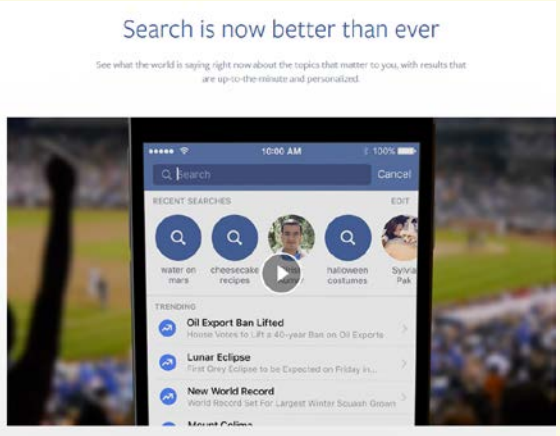


feasible for Big Data?

The bible for database researchers: Foundations of Databases
http://webdam.inria.fr/Alice/

Example queries: Facebook (Graph) Search

- ✓ Find me restaurants in New York my friends have been to in 2016
  - friend(pid1, pid2)
  - person(pid, name, city)
  - dine(pid, rid, dd, mm, yy)



- ✓ *SQL query*

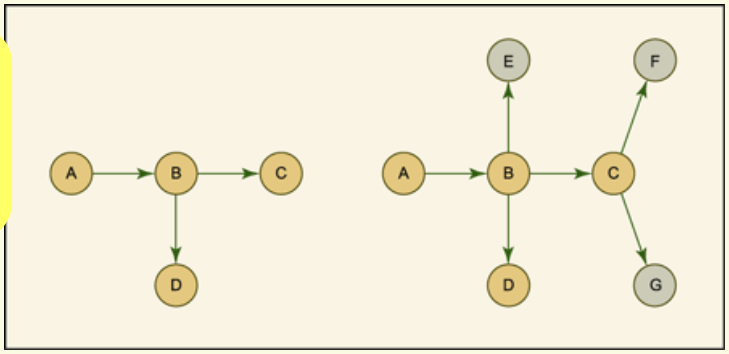
```
select   rid
   from     friend(pid1, pid2), person(pid, name, city),
            dine(pid, rid, dd, mm, yy)
   where   pid1 = p0  and  pid2 = person.pid   and
            pid2 = dine.pid and city = NYC  and yy = 2016
```

Is it feasible on big data?

Facebook : more than 1.7 billion nodes, and over 140 billion links

Example queries: Graph pattern matching

- Input: A pattern graph Q and a graph G
- Output: All the matches of Q in G, i.e., all subgraphs of G that are isomorphic to Q
- **a bijective function *f* on nodes:**
  $(u,u') \in Q$ iff $(f(u), f(u')) \in G$

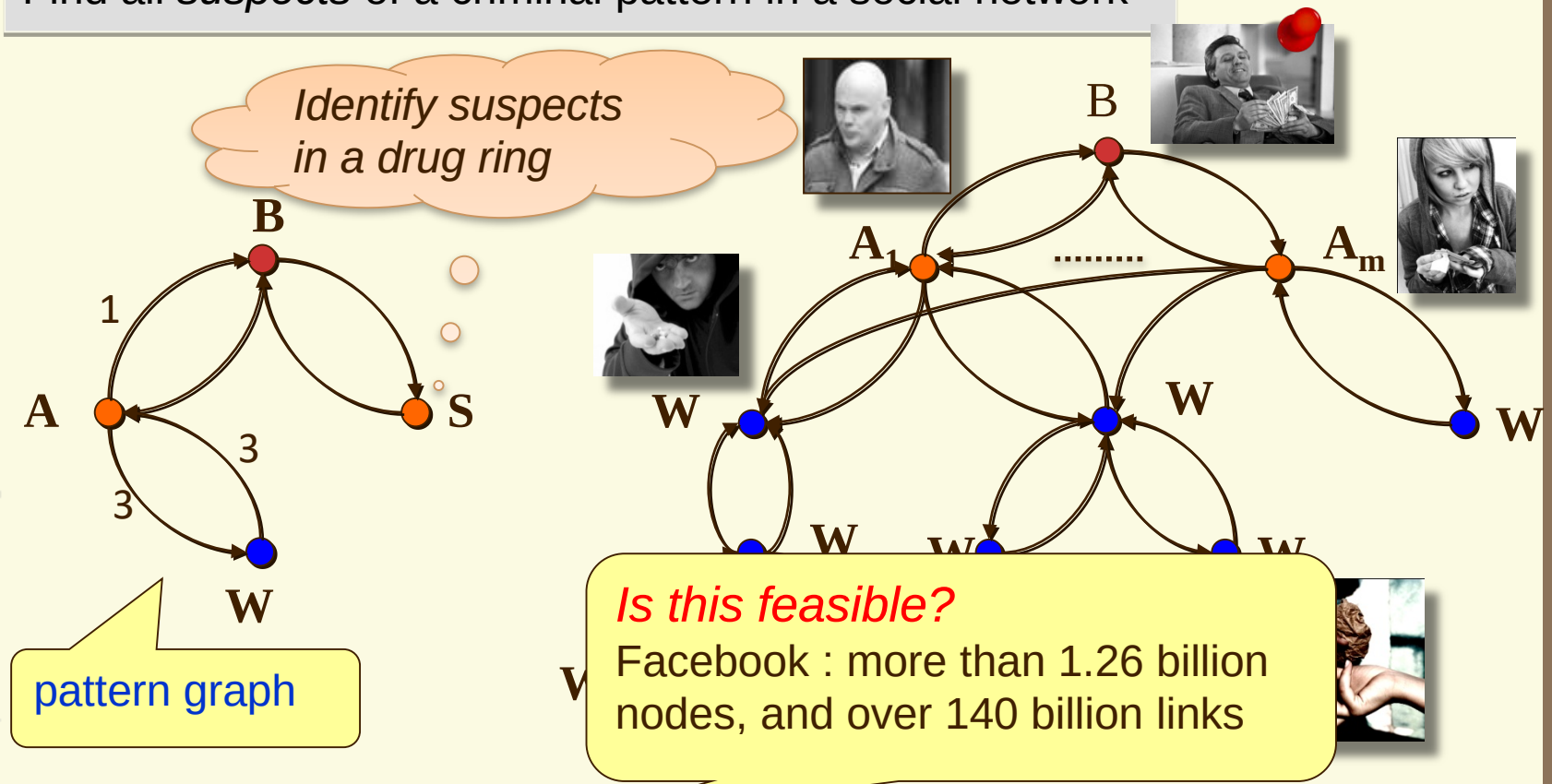


- intelligence analysis
- transportation network analysis
- Web site classification
- social position detection
- user targeted advertising
- knowledge base disambiguation …

What other graph queries do you know?

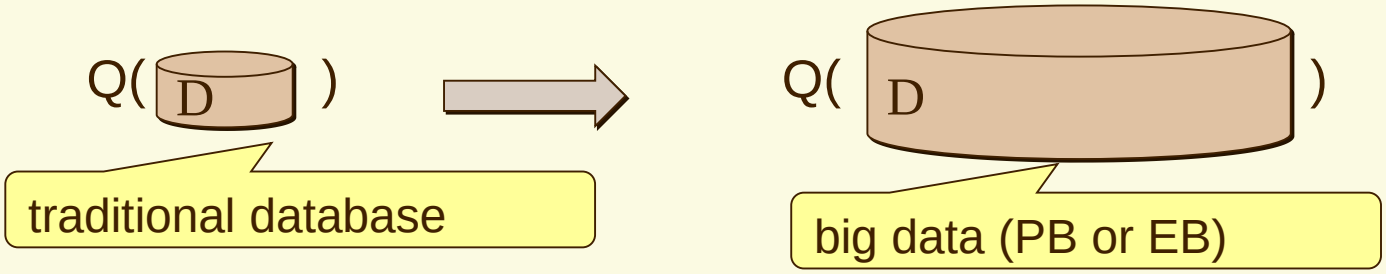
Example queries: Graph pattern matching

Find all *suspects* of a criminal pattern in a social network



Querying big data: New challenges

Given a query Q and a dataset D, compute Q(D)



What are new challenges introduced by querying big data?

- ✓ *Does querying big data introduce new fundamental problems?*
- ✓ *What new methodology do we need to cope with the sheer size of big data D?*

Why?

A departure from classical theory and traditional techniques

Classic complexity theory

Is an O(D) algorithm efficient?

How about O(D log D)?

$O(D^2)$?

$O(D^{10})$?

$O(D^{\log D})$?

$O(2^D)$?

O(D!) ?

…

polynomial time

$O(n^c)$ for some constant c

feasible, PTIME

non-polynomial Time

"intractable", NP-hard

The good, the bad and the ugly

- ✓ Traditional computational complexity theory of almost 50 years:
  - The good: polynomial time computable (PTIME)
  - The bad: NP-hard (intractable)
  - The ugly: PSPACE-hard, EXPTIME-hard, undecidable…

What happens whe

How long does it take?

Using SSD of 6G/s, a linear scan of a data set ***D*** would take

- 1.9 days when ***D*** is of 1PB (10

What query is this?

- 5.28 years when ***D*** is of 1EB (10^{18}B)

O(n) time is already beyond reach on big data in practice!

Polynomial time queries become intractable on big data!

Challenges: information extraction is costly

- ✓ Graph pattern matching by subgraph isomorphism
  - NP-complete to decide whether there exists a match
  - possibly exponentially many matches
- ✓ Membership problem for relational queries
  - ✓ Input: a query Q, a database D, and a tuple t
  - ✓ Question: Is t in Q(D)?
  - NP-complete if Q is a conjunctive query (SPC)
  - PSPACE-complete if Q is in relational algebra (SQL)

intractable even in the traditional complexity theory

Already beyond reach in practice when the data is not very big

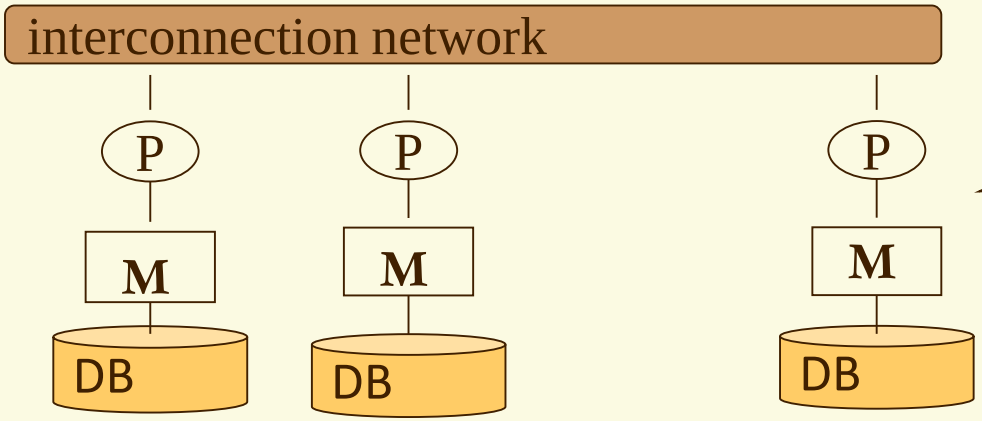
Is it still feasible to query big data?

Can we do better if we are given more resources?

- ✓ Parallel and distributed query processing

Using 10000 SSD of 6G/s, a linear scan of ***D*** might take:

- ✓ 1.9 days/10000 = 16 seconds when ***D*** is of 1PB (10^{15}B)
- ✓ 5.28 years/10000 = 4.63 days when ***D*** is of 1EB (10^{18}B)



Only ideally!

10,000 processors

Yes, parallel query processing. But how?

big data: key techniques

Big data: key techniques

Technique	Description
Data extraction, cleaning and integration	Use e.g., ETL tools to extract data from distributed, heterogeneous data sources (relational data, flat documents); clean, transform, integration, and load to data warehouse. Including real-time data stream collection and processing.
Data storage and management	Use distributed databases/file systems, data warehouse, relational database, noSQL and cloud databases to store and manage structured, semi-structured and unstructured data
Data search, analytics and visualization	Search and analyze data with distributed/parallel programming models, frameworks and systems, leveraging machine learning, data mining algorithms and tools. Visualize the result to help users understand and analyze data.
Data Security and privacy	Construct privacy protection systems and data security systems to project personal privacy and guarantee data security.

Project Overview

Project 1: Youtube Data Analyzer

- We will go through the knowledge discovery process by building a Youtube data analyzer. The process is routinely conducted in social media analytics and social network analysis.

- Goal: Implement a Youtube data analyzer supported by MapReduce, SQL and/or graph algorithms. The analyzer provides basic data analytics functions to Youtube media datasets. The analyzer provides basic search and analysis tasks.

- Dataset: http://netsg.cs.sfu.ca/youtubedata/

Project 1: Youtube Data Analyzer

- ✓ Network aggregation: efficiently report the following statistics of Youtube video network:
  - Degree distribution; Categorized statistics: frequency of videos partitioned by a search condition: categorization, size of videos, view count, etc.
- ✓ Answer user questions over the video network:
  - top k queries:
  - Range queries
  - *User identification in recommendation patterns: find all occurrence of a specified subgraph pattern connecting users and videos with specified search condition.

- ✓ Influence analysis*.
  - Use PageRank algorithms over the Youtube network to compute the scores efficiently. Intuitively, a video with high PageRank score means that the video is related to many videos in the graph, thus has a high influence. Effectively find top k most influence videos in Youtube network. Check the properties of these videos (# of views, # edges, category…). What can we find out? Present your findings.

Project 2: Airline Search Engine

- ✓ Publicly available dataset which contains the flight details of various airlines
- ✓ Goal: Implement an airline data search engine supported by efficient MapReduce, SQL/SPARQL and/or graph algorithms. The tool is able to help users to find out facts/trips with requested information/constraints:
  - Airport and airline search: Find list of Airports operating in the Country X; Find the list of Airlines having X stops; List of Airlines operating with code share; list of Active Airlines in the United States
  - Airline aggregation: Which country (or) territory has the highest number of Airports; The top k cities with most incoming/outgoing airlines

Project 2: Airline Search Engine

- Trip recommendation
  - Define a trip description language L as a regular expression $A(B)^m(\_)^nC$, where A is the source, C is the destination, and B and '_' refers to zero or more (at most m or n) stops with specified airport/city name (B) or arbitrary stop (*). For example, a trip from Seattle to Paris via London is a string: Seattle.London.Paris. Given a trip expression, develop algorithm to find the shortest possible trip. This language covers the following specific query examples you may test with.

- Define a trip as a sequence of connected route. Find a trip that connects two cities X and Y (reachability).
- Find a trip that connects X and Y with less than Z stops (constrained reachability).
- Find all the cities reachable within d hops of a city (bounded reachability).
- *Fast Transitive closure/connected component implemented in parallel/distributed algorithms connecting A and C.

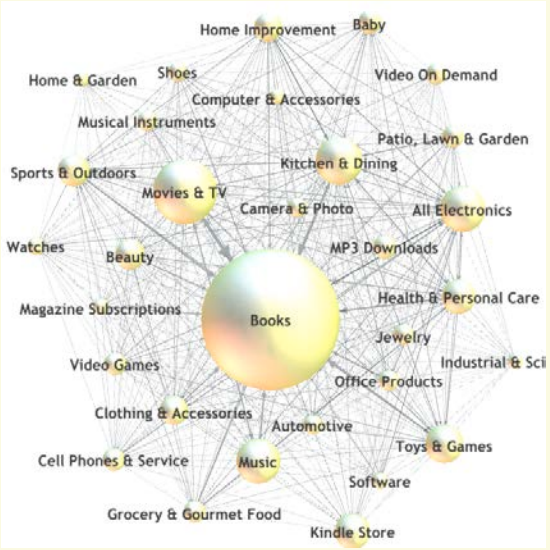
Project 3: Amazon co-purchasing pattern

- ✓ Related Industry: online commercial/Business Data: http://snap.stanford.edu/data/amazon-meta.html
- ✓ The data was collected by crawling Amazon website and contains product metadata and review information about 548,552 different products.
- ✓ Problem statement: Implement a co-purchasing data analytics engine. The analyzer has the following functions.
- ✓ Answer complex query. We define a SQL-like query Q of the form SELECT* FROM U WHERE Condition. The CONDITION is of the following forms:
  - Searchable attributes: value constraints over well defined attributes in node/edge schema
  - Non-searchable attributes: attributes that cannot be queried directly over existing attributes: the number of reviews of a product, the number of customers co-purchasing same product of a user.
  - Queries with enriched operators: >, >=, =, <, <=; e.g., Select movie with average rating >=4.5
  - Find top k entities (customers, products) for a given query Q

Project 3: Amazon co-purchasing pattern

- * Find potential customers that satisfies co-purchasing pattern. Divide the co-purchasing data into two data set, one we call “training” dataset, and the other “testing” dataset. Verify several frequent co-purchasing patterns in the training dataset. Report the frequency in the testing dataset.

- For those frequent patterns in both dataset, return the customers captured by the patterns. What seems to be the most significant co-purchasing pattern? Report your discoveries.



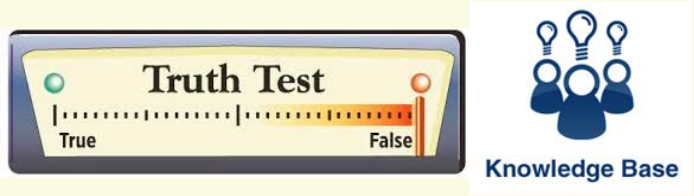
Project 4 and 5

- **Project 4: Collaboration Analysis**
  - How to discover interesting (potential) collaboration among scientists and researchers? How to track the “strong” collaboration among authors and find out the reason why some collaboration success and some is not trending? An academic collaboration analyzer gives possible answers.

- **Project 5: Offshore Leak Data**
  - A representation of Panama Papers into a network of person, companies, relationships and timestamps. 2. Mining frequent graph patterns that suggests interesting activities, potential anomaly events and outliers over snapshots of the Panama paper network. 3*. Develop algorithms that track the information of these patterns (e.g., close/reopen of companies, tax heaven, abnormal shutdown of companies, social network of users and supervisions).

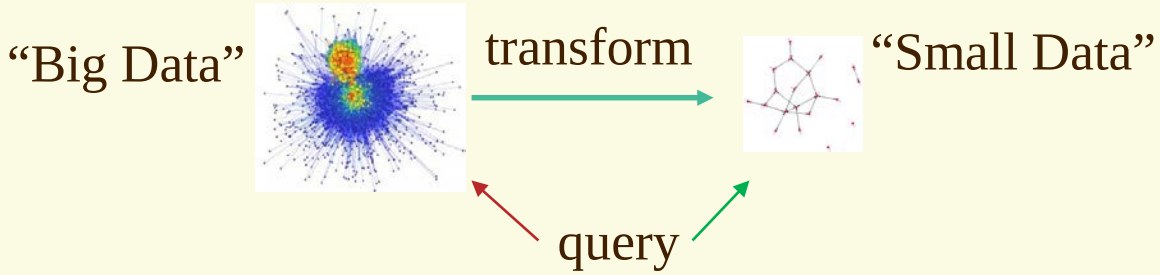
Projects: suggested for graduate students

- ✓ Mini-research projects
- ✓ **Project 6*: Knowledge base fact checker**



- ✓ **Project 7*: Knowledge base search engine**

- ✓ **Project 8*: Making "Big Data" Small**



Project Milestones

- ✓ **M1**. Select/Propose a project and understand the dataset
- ✓ **M2.** Prepare data collection and formatting. Description of data collection and the tools you use. Formulate problems; related work
- ✓ **M3. Algorithm design.**
  - Design a sequential algorithm for your problem. Description of mathematical background and data statistics from your dataset
  - Design a parallel/distributed counterpart of your algorithm using MapReduce/Hadoop or your selection of parallel/distributed computation model. Or design nontrivial optimization techniques using the Big Data algorithm design strategies (sketching, compression, sampling, indexing, views, sparsification…)
- ✓ **M4**. Experimental design and plan.
- ✓ **M5.** Experimental study/Demo and justify the result with baseline methods. Project report writing up.

Summary and review

- ✓ What are commonly seen data types and their properties?
- ✓ knowledge discovery process, and related Big data techniques.
- ✓ Understand the project tasks, milestones, and download dataset. Make a reasonable timetable for the course project and form the team.
- ✓ (next lecture) The root of data storage and management – relational DBMS and algebra

Before the next lecture

- Complete reading list RL1 below for Relational algebra and DBMS
- Get familiar with the installation of open source systems
- Think about course projects, timetable and team

Reading list 1:

- Database Systems – The complete Book: Chap 2.1-2.3, 2.5
- Database concepts: Database Management Systems, 2nd edition, R. Ramakrishnan and J. Gehrke, Chapter 1&3. http://eecs.wsu.edu/~yinghui/mat/courses/fall%202016/resources/Database%20Management%20Systems%203Rd%20Edition.pdf

*About relational databases theory and algorithms:

- Foundations of databases, S. Abiteboul, R. Hull, V. VIanu

*About big data query processing:

- Querying Big Data: Theory and Practice, W. Fan and J. Huai. JCST 2014 http://homepages.inf.ed.ac.uk/wenfei/papers/JCST14.pdf