Big Data Analytics Building Blocks; Simple Data Storage (SQLite)

Duen Horng (Polo) Chau
Georgia Tech

CSE6242 / CX4242
Aug 21, 2014

Partly based on materials by Professors Guy Lebanon, Jeffrey Heer, John Stasko, Christos Faloutsos
What is **Data & Visual Analytics**?

No formal definition!

**Polo’s definition:**
the *interdisciplinary* science of combining *computation techniques* and *interactive visualization* to transform and model data to aid *discovery, decision making, etc.*
What are the “ingredients”?

Need to worry (a lot) about storage, complex system design, scalability of algorithms, visualization techniques, etc.

Used to be “simpler” before the big data era (why?).
What is **big data**? Why care?

- **Many companies’** businesses are based on big data (Google, Facebook, Amazon, Apple, Symantec, LinkedIn, and many more)

- **Web search**
  - Rank webpages (PageRank algorithm)
  - Predict what you’re going to type

- **Advertisement** (e.g., on Facebook)
  - Infer users’ interest; show relevant ads
  - Infer what you like, based on what your friends like

- **Recommendation systems** (e.g., Netflix, Pandora, Amazon)
- Online education
- Health IT: patient records (EMR)
- Bio and Chemical modeling:
- Finance
- Cybersecurity
- Internet of Things (IoT)
Good news! Many big data jobs

- What jobs are hot?
- “Data scientist”
- Emphasize breadth of knowledge
- This course helps you learn some important skills
Big data analytics process and building blocks
Collection
Cleaning
Integration
Analysis
Visualization
Presentation
Dissemination
Building blocks, not “steps”

- Can skip some
- Can go back (two-way street)
- Examples
  - Data types inform visualization design
  - Data informs choice of algorithms
  - Visualization informs data cleaning (dirty data)
  - Visualization informs algorithm design (user finds that results don’t make sense)
How big data affects the process?

- The 4V of big data (now 5V: Value)
  - **Volume**: “billions”, “petabytes” are common
  - **Velocity**: think Twitter, fraud detection, etc.
  - **Variety**: text (webpages), video (e.g., youtube), etc.
  - **Veracity**: uncertainty of data

Schedule

- Collection
- Cleaning
- Integration
- Analysis
- Visualization
- Presentation
- Dissemination
Two analytics examples
NetProbe:
Fraud Detection in Online Auction
WWW 2007

Find **bad sellers** *(fraudsters)* on eBay who don’t deliver their items

Auction fraud is **#3** online crime in 2010

source: www.ic3.gov
NetProbe: Key Ideas

- Fraudsters fabricate their reputation by “trading” with their accomplices
- Fake transactions form near bipartite cores
- How to detect them?
NetProbe: Key Ideas

Use Belief Propagation

Darker means more likely
NetProbe: Main Results
Inspect user alisher for suspicious networks.

alisher

Registration: Aug 13, 2006
Location: United States

Suspected fraudster -- this user has been behaving much like the other suspects by trading with the similar sets of possible accomplices.
What analytics process does **NetProbe** go through?

- **Collection**: Scraping (built a “scraper”/“crawler”)
- **Cleaning**
- **Integration**
- **Analysis**: Design detection algorithm
- **Visualization**
- **Presentation**: Paper, talks, lectures
- **Dissemination**: Not released
Discovr movie app
What analytics process would you go through to build the app?

- **Collection**: IMDB, Rotten tomatoes, youtube
- **Cleaning**: May have duplicate trailers
- **Integration**: 
- **Analysis**: Determine which movies are related
- **Visualization**: 
- **Presentation**: 
- **Dissemination**: Mac app, iOS app
Homework 1 (out next week)

- Simple “End-to-end” analysis
- Collect data from Rotten Tomatoes (using API)
  - Movies (Actors, directors, related movies, etc.)
  - Store in SQLite database
- Transform data to movie-movie network
- Analyze, using SQL queries (e.g., create graph’s degree distribution)
  - Visualize, using Gephi
- Describe your discoveries
Data Collection, Simple Storage (SQLite) & Cleaning
Today:
Data Collection, Simple Storage (SQLite) & Cleaning

How to get data?

Download (where?)

API

Scrape/Crawl, or from equipment (e.g., sensors)

Low effort

High effort
Data you can just download

Yahoo Finance (csv)
StackOverflow (xml)
Yahoo Music (KDD cup)
Atlanta crime data (csv)
Soccer statistics
Data via API

CrunchBase (database about companies) - JSON
Twitter
Last.fm (Pandora has API?)
Flickr
Facebook
Rotten Tomatoes
iTunes
Data that needs scraping

Amazon (reviews, product info)

ESPN

Google Scholar

(eBay?)
Most popular embedded database in the world

iPhone (iOS), Android, Chrome (browsers), Mac, etc.

Self-contained: one file contains data + schema

Serverless: database right on your computer

Zero-configuration: no need to set up!

http://www.sqlite.org
http://www.sqlite.org/different.html
How does it work?

>sqlite3 database.db

sqlite> create table student(ssn integer, name text);

sqlite> .schema

CREATE TABLE student(ssn integer, name text);
How does it work?

insert into student values(111, "Smith");
insert into student values(222, "Johnson");
insert into student values(333, "Obama");
select * from student;

<table>
<thead>
<tr>
<th>ssn</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>Smith</td>
</tr>
<tr>
<td>222</td>
<td>Johnson</td>
</tr>
<tr>
<td>333</td>
<td>Obama</td>
</tr>
</tbody>
</table>
create table takes
(ssn integer, course_id integer, grade integer);
How does it work?

More than one tables - \textbf{joins}

E.g., create roster for this course

<table>
<thead>
<tr>
<th>ssn</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>Smith</td>
</tr>
<tr>
<td>222</td>
<td>Johnson</td>
</tr>
<tr>
<td>333</td>
<td>Obama</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ssn</th>
<th>course_id</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>6242</td>
<td>100</td>
</tr>
<tr>
<td>222</td>
<td>6242</td>
<td>90</td>
</tr>
<tr>
<td>222</td>
<td>4000</td>
<td>80</td>
</tr>
</tbody>
</table>
select name from student, takes
where student.ssn = takes.ssn and
takes.course_id = 6242;

<table>
<thead>
<tr>
<th>ssn</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>Smith</td>
</tr>
<tr>
<td>222</td>
<td>Johnson</td>
</tr>
<tr>
<td>333</td>
<td>Obama</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ssn</th>
<th>course_id</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>6242</td>
<td>100</td>
</tr>
<tr>
<td>222</td>
<td>6242</td>
<td>90</td>
</tr>
<tr>
<td>222</td>
<td>4000</td>
<td>80</td>
</tr>
</tbody>
</table>
SQL General Form

```sql
select a1, a2, ... an
from t1, t2, ... tm
where predicate
[order by ....]
[group by ...]
[having ...]
```
Find ssn and GPA for each student

select ssn, avg(grade)  
from takes  
group by ssn;

<table>
<thead>
<tr>
<th>ssn</th>
<th>course_id</th>
<th>grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>6242</td>
<td>100</td>
</tr>
<tr>
<td>222</td>
<td>6242</td>
<td>90</td>
</tr>
<tr>
<td>222</td>
<td>4000</td>
<td>80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ssn</th>
<th>avg(grade)</th>
</tr>
</thead>
<tbody>
<tr>
<td>111</td>
<td>100</td>
</tr>
<tr>
<td>222</td>
<td>85</td>
</tr>
</tbody>
</table>
What if slow?

Build an **index** to speed things up. SQLite’s indices use **B-tree** data structure. O(logN) speed for adding/finding/deleting an item

```
create index student_ssn_index on student(ssn);
```
Homework 1

Write simple scripts to import Rotten Tomatoes data into SQLite, and do some simple queries.

http://developer.rottentomatoes.com/docs/read/json/v10/Movie_Info