Data Collection, Simple Storage (SQLite) & Cleaning

Duen Horng (Polo) Chau

Georgia Tech

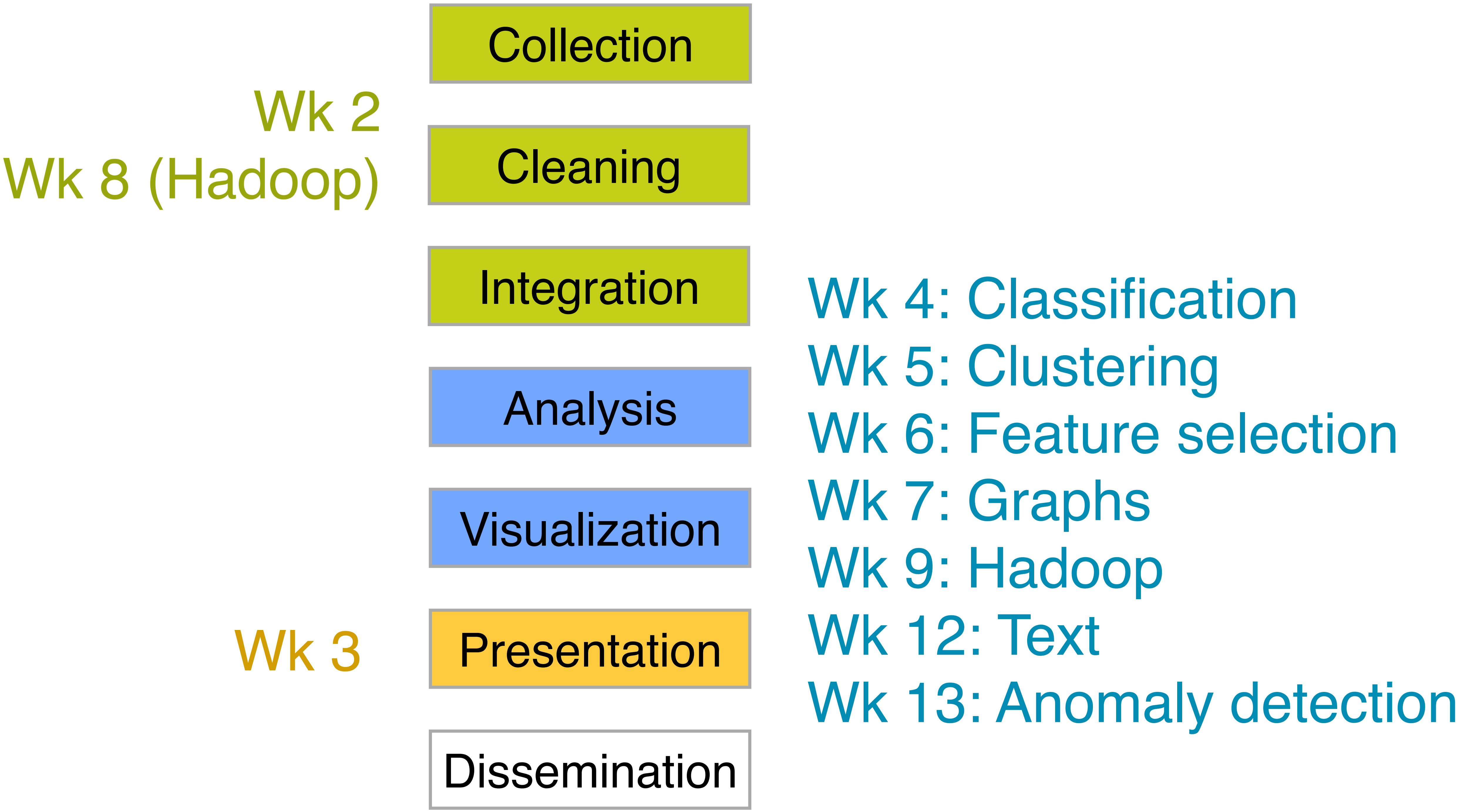
CSE 6242 A / CS 4803 DVA

Jan 15, 2013

Partly based on materials by
Professors Guy Lebanon, Jeffrey Heer, John Stasko, Christos Faloutsos

Last time:

Big data analytics process & building blocks



Collection

Wk 2
Wk 8 (Hadoop)

Cleaning

Integration

Analysis

Visualization

Wk 3

Presentation

Dissemination

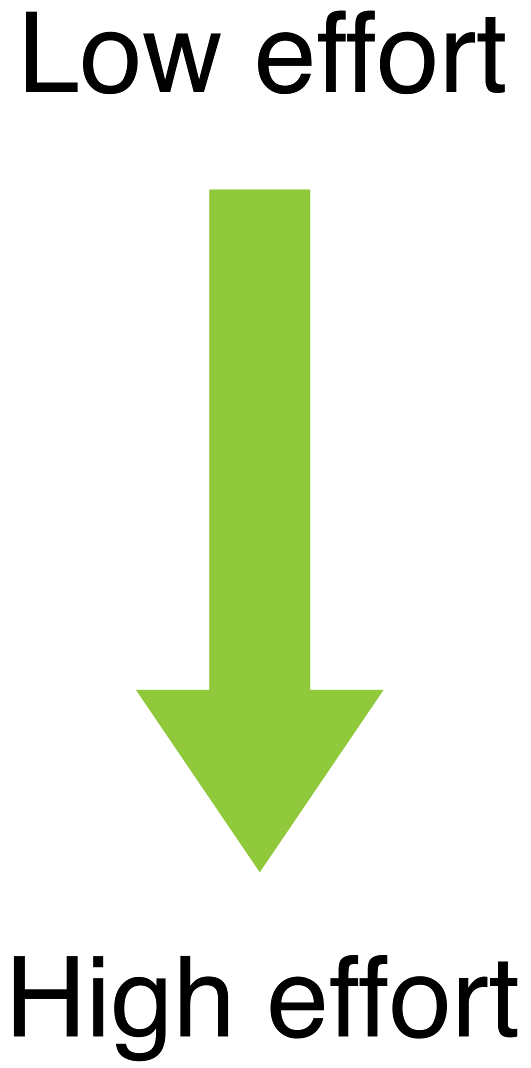
Wk 4: Classification
Wk 5: Clustering
Wk 6: Feature selection
Wk 7: Graphs
Wk 9: Hadoop
Wk 12: Text
Wk 13: Anomaly detection

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);
```

ssn	name

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;
```

ssn	name
111	Smith
222	Johnson
333	Obama

How does it work?

```
create table takes
(ssn integer, course_id integer, grade integer);
```

ssn	course_id	grade

How does it work?

More than one tables - **joins**

E.g., create roster for this course

ssn	name
111	Smith
222	Johnson
333	Obama

ssn	course_id	grade
111	6242	100
222	6242	90
222	4000	80

How does it work?

```
select name from student, takes
where student.ssn = takes.ssn and
takes.course_id = 6242;
```

ssn	name
111	**Smith**
222	**Johnson**
333	Obama

ssn	course_id	grade
111	6242	100
222	6242	90
222	4000	80

SQL General Form

```
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;
```

ssn	course_id	grade
111	6242	100
222	6242	90
222	4000	80

ssn	avg(grade)
111	**100**
222	**85**

What if slow?

Build an **index** to speed things up.
SQLite implements **B-tree**.
Speed improves from O(N) if to do a sequential scan to O(logN) .

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

Homework 1

Write a simple script/program to import Rotten Tomatoes data into SQLite, and do some simple queries.

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

How dirty is real data?

Data Cleaners

Watch videos

- Google Refine
- Data Wrangler (research at Stanford)

in Alabama	Alabama
in Alaska	Alaska
in Arizona	Arizona
in Arkansas	Arkansas

Write down

- Examples of **data dirtiness**
- Tool’s **features** demo-ed (or that you like)

Will collectively summarize similarities and differences afterwards

Google Refine: http://code.google.com/p/google-refine/
Data Wrangler: http://vis.stanford.edu/wrangler/

How dirty is real data?

Examples

- typos (missing “s”)
- inconsistency (structure)
- differences in units (billions vs thousands)
- missing values
- different values for the same thing (e.g., abbreviations)
- whole data file not in tabular format
- mixed value format (nominal vs numeric)
- different encodings
- negative values (e.g., -1 => non-sense)
- outlier in general
- different “language” (utf-8 vs ascii)
- id sometimes means a person, or household (e.g., in banking)

How do they compare?

Similarities

- work directly on data
- provide visual feedback
- browser-based
- can only hangle common use cases(?)
- free!!!
- undo/redo, history (people make mistakes)
- input: plain text

G = Google Refine
W = Data wrangler

How do they compare?

Differences

- W generates transform code
- G recognizes clusters
- W gives natural language suggestions
- G works offline (your sensitive data stay with you)
- G has more sophisticated functions?
- W seems to be able to transform overall data format
- W supports expression syntax (e.g., log())
- G more scalable(?)

G = Google Refine
W = Data wrangler

!

The videos only show *some* of the tools' features. Try them out.

Google Refine: http://code.google.com/p/google-refine/
Data Wrangler: http://vis.stanford.edu/wrangler/

Piazza

Saw some questions and answers already. Good!

- Any questions are fair game
- Questions about lectures, homework, project, tools, libraries, etc.

Has features that help form teams (for project)