Data Scraping

# Been there, scraped that

Amit Sharma, Chenhao Tan

# Why do you want to scrape data?

- It is cool to have some interesting data lying around
- Do research
  - Is there a clear question in mind?
  - What kind of data is needed?
  - What degree of comprehensiveness is needed?

# How do we scrape data?

- Processed datasets
  - Stackoverflow, Wikipedia
- Small static websites
  - Debate.org
- Large modern websites
  - Application programming interface (API)

# ***Application programming*** interface

- It is NOT for data scraping
- Respect rate limit (of course, this is my view)
  - Check rate limit
  - Add sleep between API calls
- Save all the raw data, disk is cheap, API calls are expensive

# Case study: Twitter

- Started with search API
- Search change.org and other petition sites

# Case study: Twitter

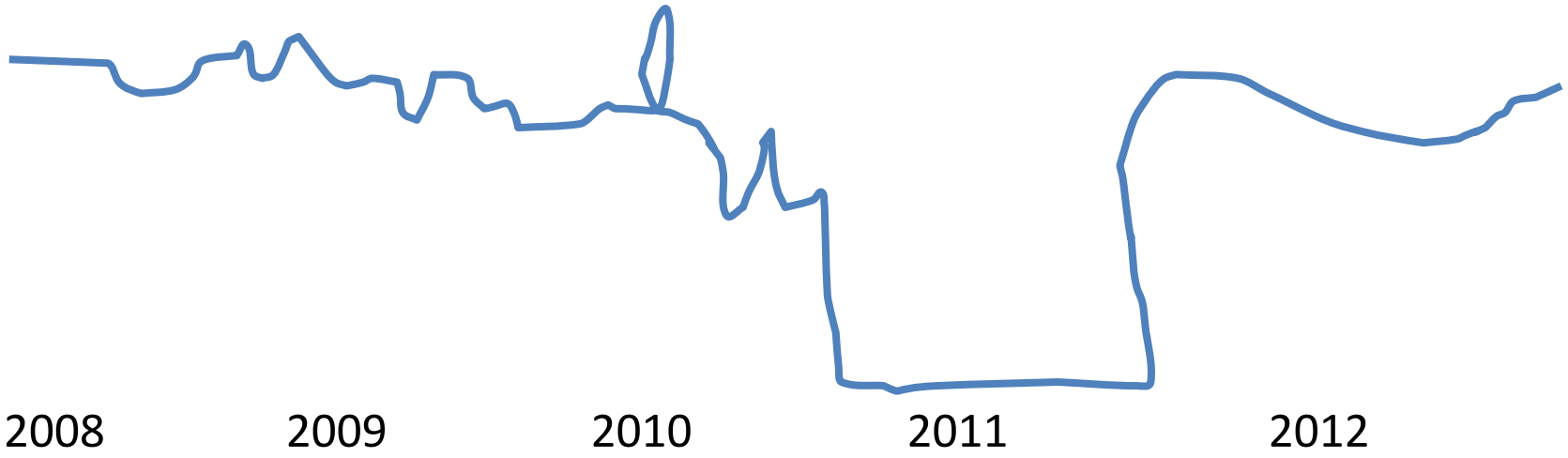
- Set up the scraping in a way that is easy to restart (keep logs, set up some ordering)
  - Switched to the user view
  - Get the most popular users from another dataset
  - Get all the tweets from those users following an order

# Case study: reddit

- The internet is your friend
  - http://www.redditanalytics.com/
  - http://www.reddit.com/r/redditdev/comments/1hpicu/whats_this_syntaxcloudsearch_do/

# Case study: reddit

- Sanity check and baby sitting



# Case study: Last.fm

1. Research Question: How do preferences evolve in social networks?

   Effects of social influences, homophily and other processes.

2. Is there a dataset already? Search, search…
3. What data attributes do I need?

   Timestamped activity data, exposure data and friendship data.

Last.fm provides all but one : timestamped listening data, love data but snapshot-only friendship data

# Case study: Last.fm

## Biases, biases, biases…

Your sampling strategy will create biases.

Your research question will guide which biases to nurture ( e.g. inactive users are not useful for studying temporal preferences, but critical for studying why users leave)

I needed information on friends for each user, and also a reasonably connected component.

So chose weighted BFS

# Case study: Last.fm

How much data do you need?

---parallel programming

--I first wanted to  implement parallel BFS (!).

Data will never be perfect

-- robust error checking (RTFM!), email scripts

Think hard about data format

-- flat files, databases, json?

Contributions

-- data, code (why not a general library for data crawl?)

# Our version of summary

- Think about what data you need
- Search for tips/existing solutions
- Start with small, manageable size, at least estimate how long it may take
- Keep logs and the raw data
- Sanity check and baby sitting