IS5126 – HowBA
Lecture 2 – Data, collection, and web scraping
Jan 19, 2015
Dr. Tuan Q Phan
NUS IS5126

Learning Objectives
• Data sources – Primary vs. secondary/3rd party
• Scraping Basics
• Python Programming
• Parsing HTML

Admin
• Syllabus and schedule on IVLE and homepage, http://www.tuanphan.us
• Purchase HBS Case from http://hbsp.harvard.edu
  – Data.gov, #9-610-075
• Sign up team of 4 on IVLE by Jan. 30
  – Use IVLE forums to find team mates

Data & Data Collection

Primary Data Sources
• Devices
  – Point of sales (POS)
  – Kiosk
• Infrastructures & Systems
  – Websites (web logs, cookies, online payment)
  – Electronic Resource Management (ERM), Customer Relationship Management (CRM), Human Resource (HR) systems
• Products & Services
  – Mobile devices (usage, location, transactions)
  – Cloud services

Secondary/3rd Party Sources
Public Data
• Data.gov (BLS, economic, census, CDC, FDA, FEMA, geo, trade, weather, etc...)
• Data.gov.sg
• States/provincial governments
  – California Department of Education: http://www.cde.ca.gov/dt/
• National-level:
• International – UN, WHO
• Universities/Research/Academics
  – Princeton University:
  – UCI Machine Learning:
    http://archive.ics.uci.edu/ml
  – UCR Time-series:
    http://www.cs.ucr.edu/~eamonn/time_series_data/
Secondary/3\textsuperscript{rd} Party Sources
Direct/Commercial Data & Clearinghouses

- Thomson Reuter (News, business)
- Forrester Research (business, surveys)
- NYSE, Yahoo Finance
- Nielsen Data (entertainment, social media)
- Sports-reference.com

Secondary/3\textsuperscript{rd} Party Sources
APIs

- Google API (maps, youtube, search, trends, AdSense)
  - Yahoo (search, image, BOSS, local, ...)
  - Bing
- eCommerce: Amazon, ebay, Yahoo,
- Social Media API: Facebook, twitter, linkedin
- Entertainment: IMDB, last.fm, LyricWiki, netflix
- Real estate: zillow
- Aggregators: ~7000 API, ~7000 “Mashups”
  http://www.programmableweb.com/

Secondary/3\textsuperscript{rd} Party Sources
Scraping

- When data is not easily available
- Used to get competitor’s information
  - Google search engine (robots)
  - Amazon’s price comparison
  - ITA (Google/Orbitz) flight inventory
  - Nielsen data
  - Baidu/Tao Bao/Yi Hao Dian/360
- Can be blocked

Scraping Basics
Web Technologies

- HTTP:
  - Specifies communication between client (browser, robots) and server
    - HTTP Pocket Reference: Hypertext Transfer Protocol, Clinton Wong
  - URL as global address
  - “GET” a web document, e.g.
    http://www.nus.edu.sg/index.html
  - “POST” sends data to the server, e.g. Forms, login, photos
  - Server responds with content (e.g. HTML)

Scraping Basics
Approaches

- Excel:
  - Built-in feature for simple extraction (Windows only)
  - Pros: GUI, easy to get started
  - Cons: limited, not scalable, windows-only
- Selenium:
  - debugging tool with client/server architecture
  - Integration with browser (Firefox)
  - Scripts to click, extract, and interact with webpages
  - Pros: easy to start, can interact with complicated (flash, AJAX) websites, less sensitive to changes
  - Cons: slow, limited, scalability
Scraping Basics

Approaches

- Robots
  - Small piece software to extract content from webpages
  - Pros: highly scalable to multiple simultaneous connections, fast, robust
  - Cons: larger initial hurdle, sensitive to webpage changes, tricky to handle complicated sites (AJAX, Flash)
- Frameworks (e.g. Scrapy)
  - Software packages to ease writing robots
  - Pros: incorporates best practice, highly scalable, tried and tested, fast development
  - Cons: larger overhead to learn, performance issues
- Other:
  - Yahoo Pipes
  - Google scripting

Scraper Structure

1. Grab webpage at entry URL
2. Parse HTML, extract & store data
3. Follow children links, grab children pages
4. Parse HTML, extract & store data
5. Repeat steps 3-5 as necessary

HTML in 1 slide

- “Webpage”
- Hyper Text Markup Language (HTML)
- Text file which tells your browser how to display content (text, images, videos, etc…)
- Tags: <html> ... </html>
- Rich media, attribute src: <img src="me.jpg" />
- Links: <a href="nextpage.html">click here</a>
- Tool: Firebug Plugin for Firefox
- Resource: http://www.w3schools.com/html/default.asp
Python (2.x) Programming Basics

- Easy to learn, powerful and versatile
- Can run on any platform
  - Cygwin for Windows environment recommended
- Popular in production systems (eg. Google)
  - But also good for BA and prototyping
- Text file as a program
- Any plain text editor
  - no Word, Notepad, Textedit, Notepad++, Sublime

Python (2.x) Programming Basics

- Command line: python myscript.py
- Run interactive: python
- Wide set of libraries that extends functionality
  - (eg. BeautifulSoup4)
- Package managers (PIP, easy_install, Debian, etc...)
- Other packages of interest:
  - NumPy, SciPy, NLTK
- Resource: http://docs.python.org/tutorial/

Python (2.x) Programming Basics

- Each line (ends with a NEWLINE) is executed sequentially
- # denotes comment, not executed
- Spaces and tabs matter!
- Call functions: myvar = myfunc(parameter1, parameter2)
- Variables contain values, primitive types:
  - Numbers: int, float, long: Operators: +, -, *, /
  - Strings: characters, immutable. Operators: +
  - Lists: mixed types, immutable. Operators: [], len()

Python Programming

- Reading & Writing Files
  - Read a file, line by line
    f = open("/path/to/file.txt","r")
    for line in f:
      print line # prints every line of a file
    f.close()
  - Write a file
    f = open("/path/to/writefile.txt","w")
    f.write("Content of my file, and some more")
    f.close()

Python Programming

- Conditionals & Loops
  - Conditional:
    a = 5
    if a > 1:
      print "a is greater than 1"
    else:
      print "a is not greater than 1"
  - Loops:
    for i in range(10):
      print "i value = " + str(i)  # i value = 0; i value = 1; ...

Python Programming

- Parsing HTML Strategy
  - “dumb” text search and extraction (regular expressions, aka. Regexp)
    - Uses powerful “mini language” to search for text
    - Pros: flexible, powerful tool, can handle mal-formed HTML code
    - Cons: sensitive to changes, easy to make mistakes
  - HTML parsing engine (BeautifulSoup4)
    - Use libraries to parse HTML and tags
    - Pros: less complexity, easy to get started
    - Cons: slow, mal-formed HTML errors
  - Combined

ID: Tuan Q PHAN, NUS IS5126, (c) 2015
**Parsing HTML**

**Regex**

- Looking for a "needle" in a "haystack"
  
  ```
  haystack="abcdef"
  needle = "cd"
  m = re.search(needle, haystack)
  m.group(0)  # "cd"
  ```

- Special characters:
  - `.`: any character
  - `*`: matches 0 or more of preceding character
  - `+`: matches 1 or more
  - `?`: matches 0 or 1

  ```
  haystack = "abcdef"
  m = re.search("cd\.*", haystack)
  m.group(0)  # "cdef"
  ```

- **Groups:** extracts values from the needle.
  ```
  m = re.search("a(\bc\)d\(\.
  \))f", haystack)
  m.group(1)  # "bc"
  m.group(2)  # "f"
  ```

- **Character Classes:** specifies certain characters
  - Characters between `[` and `]`
  - "Not" character `^`
  - `cd[^f]*`:

- **Anchors:**
  - `^`: beginning of a line
  - `$`: end of a line

---

**Challenges**

- Logins and Cookies
- Optical challenge
- Rich content (AJAX, Flash, etc...)
  - Selenium
- Robot blockers/Throttling
  - Distribute robots
  - Slow scraping or "smart" behavior

---

**Admin**

- Syllabus and schedule on IVLE and homepage, [http://www.tuanqphan.us](http://www.tuanqphan.us)
- Purchase HBS Case from [http://hbsp.harvard.edu](http://hbsp.harvard.edu) – Data.gov, #9-610-075
- Sign up team of 4 on IVLE by Jan. 30 – Use IVLE forums to find team mates