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.tuanqphan.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

DATA FLOW

Real World

Collection

Raw data
Import
Data warehouse
Analyze
Report

Transform

Primary
- Devices
- Infrastructure & Systems
- Products & services

Secondary/3rd party
- Public data
- Direct/Commercial
- Data clearinghouses
- APIs
- Scraping

Dr. Tuan Q. PHAN, NUS ISS126, (c) 2015
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

Dr. Tuan Q. PHAN, NUS IS5126, (c) 2015
Secondary/3^{rd} 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/ds/
- National-level:
- International – UN, WHO
- Universities/Research/Academics
  - UCR Time-series: http://www.cs.ucr.edu/~eamonn/time_series_data/

Dr. Tuan Q. PHAN, NUS IS5126, (c) 2015
Secondary/3rd Party Sources
Direct/Commercial Data & Clearinghouses

- Thomson Reuter (News, business)
- Forrester Research (business, surveys)
- NYSE, Yahoo Finance
- Nielsen Data (entertainment, social media)
- Sports-references.com
Secondary/3rd 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/

Dr. Tuan Q. PHAN, NUS IS5126, (c) 2015
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

Dr. Tuan Q. PHAN, NUS IS5126, (c) 2015
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, eg
  – “POST” sends data to the server, eg. Forms, login, photos
  – Server responds with content (eg. HTML)
Scraping Basics
Web Technologies

• Photos
• “webpage”
• XML
• HTML
  – Flash
  – AJAX
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 (eg. 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
Break
Scraping Basics
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
Scraping Basics
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

Dr. Tuan Q. PHAN, NUS IS5126, (c) 2015
Best Sellers in Computers & Technology

1. 315 days in the top 100

Steve Jobs
Walter Isaacson
⭐⭐⭐⭐⭐ (1,538)
Kindle Edition
Auto-delivered wirelessly
Kindle Price: $14.99

2. 207 days in the top 100

Kindle Fire Tips & Tricks
Tim Sievers
⭐⭐⭐⭐ (17)
Kindle Edition
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

Dr. Tuan Q. PHAN, NUS IS5126, (c) 2015
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/](http://docs.python.org/tutorial/)
Python (2.x) Programming Basics

• Each line (ends with a NEWLINE token) is executed sequentially
• # denotes comment, not executed
• Spaces and tabs matter!
• Call functions: myvar=myfunct(parameter1, parameter2)
• Variables contain values, primitive types:
  – Numbers: int, float, long. Operators: + - * /
    a=5
    b=1
    a+b #6
  – Strings: characters, immutable. Operators: +
    first = “Tuan”
    last = “Phan”
    first + last # “TuanPhan”
  – Lists: mixed types, immutable. Operators: [], len()
    a = [‘first item’, ‘second item’, 100, 1234]
    a[2] # ‘second item’
Python (2.x) 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
Reading & Writing Files

• Read a file, line by line
  
  ```python
  f = open("/path/myfile.txt","r")
  for line in f:
    print line # prints every line of a file
  f.close()
  ```

• Write a file
  
  ```python
  f = open("/path/writefile.txt","w")
  f.write("Content of my file, and some more")
  f.close()
  ```
Parsing HTML Strategy

• “dumb” text search and extraction (regular expressions, aka. Regex)
  – 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
Parsing HTML

Regex

• Looking for a “needle” in a “haystack”

```python
import re
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

Ex:

```python
m = re.search("cd.*", haystack)
m.group(0)  # "cdef"
```
Parsing HTML

Regex

• **Groups:** extracts values from the needle. ()
  
  ```python
  m = re.search("a(bc)d(.f", haystack)
  m.group(0) # "abcdef"
  m.group(1) # "bc"
  m.group(2) # "e"
  ```

• **Character Classes:** specifies certain characters
  
  – Characters between [ ]
  
  – “Not” character ^
  
  – “cd[^f]*”

• **Anchors:**
  
  ^ beginning of a line
  
  $ end of a line
Potential issues:

- May not match on other product pages
- May match multiple needles, but not the right one!
from bs4 import BeautifulSoup
import urllib

urlHandle = urllib.urlopen("http://www.amazon.com")
html = urlHandle.read()
soup = BeautifulSoup(html)

priceLarge = soup.find_all("class","priceLarge")
for p in priceLarge:
    price = p.text
    print("$14.99")
Challenges

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