A Guide to Web Scraping For Economists*

Abhishek Nagaraj †
nagaraj@mit.edu

January 20, 2014

Abstract

NOTE : THIS IS STILL A WORK IN PROGRESS.
This article goes over the basics of how economists might go about collecting both structured and unstructured data from the internet. I will discuss basic concepts, and then discuss how to implement these ideas in python. I conclude with examples of existing research that leverages micro-data collected from the internet for interesting applications.

*Thanks to xxx. All errors remain mine.
†MIT Sloan School of Management. E62-368, 100 Main Street, Cambridge, MA 02139.

1 Introduction

The internet is a wonderful places for economists to collect data for research. As an illustration, markets like eBay and Amazon can be used to study auctions, market designers can record data on interactions in online dating, macroeconomists can study price stickiness in online supermarkets and those interested in labor markets can observe job postings, employment decisions and LinkedIn profiles.

One prominent technique to get access to such data is "web-scraping", or "screen scraping" or simply "scraping". Put simply, scraping is the process of programatically accessing digital content on the internet and parsing this often unstructured content into familiar flat-file style data frames that can then be analyzed in STATA or R. In this article, I will discuss the basics of these techniques, a few advanced suggestions that might be useful once you've mastered the essentials and highlight examples of such data being used effectively in research.

Two important caveats before I begin. First, in most cases, before you being any scraping project, you should read the Terms of Service of the website that you're trying to scrape. Most websites explicitly prohibit screen scraping of their content. For example, LinkedIn's TOS states (emphasis mine)

> ... we grant you a limited, revocable, nonexclusive, nonassignable, nonsublicenseable license and right to access the Services, through a generally available web browser, mobile device or LinkedIn authorized application (*but not through scraping, spidering, crawling or other technology or software used to access data without the express written consent of LinkedIn or its Members*),

Despite such legal challenges, in pratice, scraping seems to reside in a legal grey area and is quite common on the internet. Indeed, Google itself feeds its search engine through the use of such scraping techniques. As a practical

matter, if you decide to go ahead and scrape content, respect the bandwith of the website you're scraping viz. do not query pages repeatedly without any delay or do not access thousands of pages at once. Also be aware that violating Terms of Service is a criminal offense under the Computer Fraud and Abuse Act (1986).

Second, this article is written for those who have always wondered about the possibilities of using scraped data, but have never quite understood how, and is more of a primer, rather than a detailed manual. If you're looking for something more advanced, this is probably the wrong place. Finally, the concepts that I introduce should apply generally, but I will also discuss specific implementation strategies in python.

2 So You Want to Scrape?

So, you've decided to get into web scraping? Smart move, but my first suggestion to you is "Think Again". If you have access to structured data, data that's already been collected and processed for you, data available from websites through their Application Programming Interface (in other words, a programmatic way to access structured data) then you should choose these alternatives ahead of scraping. In practice, unless your project is very simple and self-contained, scraping is messy, never perfect and requires a fair bit of maintenance and hand-holding.

However, often the perfect dataset is not available by default, which is when having scraping in your toolkit is a wonderful addition. When scraping works, a lot of wonderful research is possible with scraped data. The following example walks you through one basic scraping project that will make a lot of the basic concepts clear.

3 Sample Project

Consider the following simple webpage.

1.

2.