Big Data for Computational Social Science
Spring 2012

Max Tsvetovat

January 24, 2012

1 Why?

Big Data is ...uhm... big. And trendy and stuff. But why should we bother?

While traditional social science research could be done with relatively small datasets – hundreds or thousands of datapoints – and these datasets were small enough to be dealt with in flat files using well-known tools like Excel, SPSS, etc. However, the size of social data was limited not by what was really needed to do the research, but rather by cost and difficulty of data acquisition. If all one has to rely on is interviews, hand-administered surveys, and manual text analysis, collection of a few hundred data points could easily spiral into a multi-year project.

So here comes the Web – and, even more so, the Social Web. Blogs, Wikipedia, Twitter, Facebook, all fundamentally changed the nature of social discourse – and the nature of social science research. Now it is actually possible to administer a social-science experiment to millions of respondents (see Facebook's "I Voted" experiment), or collect 500 million tweets in a few months of relatively simple data collection regimen. Computational social science and computational models can generate more structured data – but can easily top these volumes as well.

When one crosses the barrier between "small data" ($< 10,000$ data points) to "medium data" ($10^5 - 10^6$ data points) the transition is fairly straightforward and well-understood. Get an SQL database, and you're in business. However, crossover to Big Data ($> 10^7$ data points) presents a completely new and different problem.

The good news is that the tools for Big Data computation are now fairly mature. The bad news is – relatively few people know how to use them (in comparison to Excel or SQL users, that is).

This class is an experiment. There are only a few courses that teach cloud and Big Data computing in the country – I counted 5 or 6. I will learn as much or more then you in this class. Let's make it work!

2 What?

What are we going to do? How are we going to learn Big Data computing?

We are going to hack Wikipedia. Or, rather, learn everything we could possibly learn about the Wikipedia data, and do some useful research with it.

Wikipedia provides the entire corpus of data to researchers, and I have downloaded all 7 GB of it to an S3 volume on Amazon. We'll hit it with simple linguistic analysis, network analysis, change statistics, machine learning – whatever we can muster. I will provide a small corpus for quick testing, but eventually we'll hit the whole thing.

3 When?

We meet on Tuesdays at 4:30 in Innovation Hall. My office hours are on Tuesday and Thursday afternoons (2pm - 4pm) in Research 1 / rm 381 or by appointment.

3.1 Twitter

My professional communications are all on Twitter, **maksim2042** (if you want to know what jazz record I'm listening to then try Facebook ;-). I will tweet things related to the class using a hashtag **#bigsocialdata**, and I encourage you to tweet interesting articles using the same hashtag. Follow me and I'll follow you ;)

3.2 Computing

This class is sponsored by Amazon Academic Computing Grant – we have way more computing power to play with then we need. I will set up authen-

tication so everyone can start machines and clusters under my sponsored Amazon account.

If you bring your laptop, it'll make our lives easier – but the workstations in the class are sufficient for the purpose (i.e. opening an SSH session to the Amazon cluster).

Programming languages for the class are Java and Python – if you must use R or C# or whatever, you're on your own. If you're not a proficient programmer, you will suffer in this class. Sorry.

3.3 Books

We get our book fix from O'Reilly. As an author, I have a huge discount so we can order in bulk on the first day of class.

- Hadoop: The Definitive Guide – By Tom White
- Programming Pig – by Alan Gates [1]
- Programming Collective Intelligence – by .....
- Online tutorials for Hive, Cloud9 and other tools

3.4 GitHub

There is a GitHub repository for this class; I will use it to post papers, slides and code. You will use it to submit code to our joint open-source effort (i.e. the whole class).

https://www.github.com/maksim2042/BigDataClass

3.5 Grading

You will get an "A".

Unless you miss a bunch of class sessions (for no good reason) and do very little of the course work. Or do something obnoxious. Or otherwise disrupt or disregard the intellectual climate that I'm trying to build in this class.

If you can't get stuff to work, we'll take extra time to make sure that we figure it out. This is a new subject, there's really no point in doing this

[1] no, this is not about a swine developing iPhone apps. Pig is a nice Big Data analysis tool

class other then to learn. The schedule (below) is intense, we'll all lose some sleep over it (yours truly included), and I have plenty of slack in it to take up anything that blows up.

4 The Plan

Week 01 : What is cloud computing – what is Big Data – why is this different – what is Hadoop – setting up a Hadoop cluster

Reading: the Map-Reduce paper (on GitHub)

Work:

(a) set up a Hadoop cluster

(b) using MapReduce, compute word count in Wikipedia dataset.

Week 02 : MapReduce

Reading: really, read the MapReduce paper. I mean it.

Work: (can be done using Java + Cloud9 library, or Python + Boto library, your choice. I'll use Python)

(a) using MapReduce, compute word frequency distribution in Wikipedia articles.

(b) using MapReduce, compute and analyze bigram counts

Week 03 : Indexing

Indexing is an important mechanism for being able to find things later on. Many tools for successful indexing exist (Lucene, SOLR, etc) – but we're going to build a simple indexer on our own just to see how they work. Our indexing method will suck compared to these offered in off-the-shelf tools but we'll learn something in process.

Reading: ¡¡TODO: find a good paper on inverted indexes¿¿

Work:

(a) Write an indexer using the inverted index method. Index the Wikipedia dataset. Use the index to search it

(b) Modify your indexer to index bigrams instead of words

Week 04 : Deriving Link-Graphs

Wikipedia pages link to other Wikipedia pages. This forms a huge network – which approximates the topic-graph of the human knowledge. Let's find this graph.

Reading: ¡¡TBD¿¿

Work:

(a) Write a map-reduce job that takes the corpus of Wikipedia pages and returns a link-graph of Wikipedia pages

(b) Compute degree centrality of wikipedia pages

Week 05 : PageRank

Did I tell you we're going to clone Google? ;)

Reading: The Page Rank Paper (on GitHub)

Work:

(a) Using MapReduce, compute PageRank of Wikipedia articles using the link-graph from last week

Week 06 : Querying Big Data

Now that we have derived a (large) pile of structured data from our Wikipedia corpus, it's time to do something useful with it. Choose Hive (SQL syntax) or PIG (simpler syntax). We are going to use output from all previous exercises as input for next few weeks.

Reading: Hive or Pig manual, your choice

Work: (a) Get Hive or Pig installed and get it to read the files. (b) Perform some simple queries to make sure things work; do some exercises from the manual

Week 07 : Querying Big Data

Let's see if we can learn something about what pages are prominent on Wikipedia and what makes them important. We will combine bigram index with pageRank data

Reading: Hive or Pig manual. You'll still need it.

Work: (a) Using a tool of your choice (Hive, Pig or hand-built), join the bigram index and PageRank. Come up with a bigram signature (i.e. most prominent bigrams) of pages with high page-rank

(b) Build a new index that sorts Wikipedia pages matching a query string in order of highest PageRank

(c) Congratulations – you have now replicated the 1999 version of the Google Algorithm.

Week 08 : I'm out of town at PyCon. Perhaps we'll get an invited speaker

Week 09 : Spring Break. I will be at SunBelt running a Big Data hackathon. You all are welcome to join me.

Week 10 : Big Data and Machine Learning

Let's see if we can derive topic clusters from our bigram/PageRank index.

Reading: Inhale a portion of "Collective Intelligence" book or your favorite text on clustering

Work: (a) Implement a cosine distance document clustering as a MapReduce job

(b) Think of a cool project to do for the rest of the semester.

Week 11 : Project

Let's split the class up into a couple project groups and find good things to do with our data. Things are wide-open from now on.

Reading: ¡¡TBD¿¿

Work – what are you waiting for? Get cracking on the project!!! ;)

Week 12 : Project

What did you learn in the first week of working on the project? Prepare a short talk about a new idea or algorithm that you're exploring. I will prepare a talk on whatever topics the groups will need

Week 13-14 : Project hacking. We get together for project updates, spend time brainstorming and helping each other solve problems.

Week 15 : Final presentations and paper submissions

I want you all to present at Strata if we can. So let's make this really nice!

Note: Things will go wrong. Writing code will take longer then we think (it always does). Snow will fall. Internet will get censored (God forbid). I built a 3-week slack into the course just for that

5 Collaboration

This course is one big collaborative project. Let's all be friends and help each other. Polite people say "thank you" when they receive help – and acknowledge their helpers in their work product[2]. I will follow standard academic practices in citing people, and so should you.

[2]If they didn't teach this in kindergarten...