Saturday, October 3, 2015

7 Links To Convince You That Big Data Isn't Your Problem

tl;dr - scroll to the bottom for a list of 7 links you should can read instead of this entire post.

I'll spare you the big bulk of my words and get to the point:

From my personal experience with big data projects, I'll conclude that it's a big (data) fat lie (or a big fat [data] lie if you'd like). Most people don't have big data to begin with, and they use big data technologies for the saying 'we use big data in our product' to acquire more sales. Below is the story in a set of links: read through this links, while bearing in mind the key points I attached to each one of them. I hope that by the time you finish reading this, you'll become more convinced in this point.

And yes, big data technologies still have a place in today's world: buzzwords have always been a part of business, and somebody has to purchase all of these disks and RAM and CPU (packed nicely into "commodity servers"). But if you are a penniless start-up worrying about how will you close the gap and get in the game, rest assured. The game is mostly imaginary - as always!

I dedicate this post to my friend Rony, an economist who asked me for tips on how to incorporate "big data" into her research. A short set of questions proved to me that she was not aware of what big data really is, and that it wasn't relevant for her case. She would use computers, programming and databases - but she should not worry that she is missing any knowledge. So for people like her, whose fears get fed by the same legends big data salespeople spread about it to get more cash, I dedicate this post.

The Beginning: Google Gets There First (as always!)

I mark the beginning of the era of 'big data' with two research papers published their Google.

The first paper described the Google File System (GFS) and you can (and should!) read it here:
http://static.googleusercontent.com/media/research.google.com/en//archive/gfs-sosp2003.pdf
This is a file system that specializes in scalability, in appending incoming data in mass volumes into existing files, and in making high-availability systems (due to redundancy of data). These were all valid concerns relevant for Google's business - but not relevant for just any company.

The second paper described the MapReduce paradigm:
http://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf
Basically that means that if I have a book and I want to count how many times each word appears in it - I can split it to several chunk, count the words separately in each chunk, and then combine my results. The split+count part is called **MAP**, the combine part is called **REDUCE**. This is the nuttiest of all nutshells explanations, but is enough to get some few points across. First, this paradigm fits for batch processing of rather homogenous data. Second, it doesn't fit non-aggregative batch processing tasks. An example is Graph Algorithms - looking at a node in a graph, and retrieving its friends, its friends' friends... etc - is something you don't normally do using MapReduce.

These two concepts are combined into a single system like this: the data is stored using a GFS-like filesystem, then processed using the MapReduce paradigm (the more general term is 'lambda architectures'). Google published these papers but did not release any code. But Doug Cutting and Mika Cafarella, two engineers from Yahoo, decided to implement such a system as part of their work in Yahoo. This became an open-source Apache project named **Hadoop**. So - Hadoop - open-source GFS+MapReduce. Hadoop became the generic name for a big data processing system and is used widely in academia and industry. But enough about Hadoop at this point.

You might notice that these papers are from 2003 and 2004. Google itself has already got past that point and is using some other technologies which don't have open sources implementations out yet. You can read more about how Hadoop is no longer the cutting-edge big data framework here:

http://cacm.acm.org/blogs/blog-cacm/177467-hadoop-at-a-crossroads/fulltext

The Disillusionment: Where Big Data Fails in General, and Hadoop fails Specifically

Instead of presenting my own opinion, please look at this KD Nuggets poll:
http://www.kdnuggets.com/2015/08/largest-dataset-analyzed-more-gigabytes-petabytes.html
When asked what is the largest dataset analyzed, data scientists (the people working on analyzing data) respond that a laptop is usually enough for their needs.

(I used to link the previous year's poll when I noticed a new version came out. Still, here it is, you can compare each year's results to see the stability:
http://www.kdnuggets.com/polls/2014/largest-dataset-analyzed-data-mined-2014.html
)

Why is that? For two reasons. The first one is that big data is not always necessary to develop a good data-driven application. Let's say we are trying to create personalized messages for e-commerce websites. Is the entire history of all users necessary? Maybe just the last 3 months? How much do they weight? Is a sample of the data enough for building a good model (rather than the entire corpus)? Is the entire corpus tagged (labelled) so that we can use it for building our model? This is not always the situation. Quantity can not always cover for quality.

The second reason is that most big data systems are aimed at being scalable. That means that such a system deployed on two machines (working together on the computations) will perform better than the same system deployed on a single machine (and it goes on: 3 machines are better than 2, 4 are better than 3... etc). The more machines also allow for robustness - if one of the machines die in a Hadoop cluster, the others can pick up from where it left and finish the job. So big data systems are usually scalability-oriented. Nice!

But do they perform better than a non-scalability-aimed, single-thread based system?

Frank McSherry *et. al* checked this subject and returned with surprising results. Read their paper here:
Frank McSherry, Michael Isard, Derek G. Murray
Scalability! But at what COST?
http://www.frankmcsherry.org/assets/COST.pdf

A shorter reading would be McSherry's blog-post about this paper:
http://blog.acolyer.org/2015/06/05/scalability-but-at-what-cost/

And also his follow-up post about testing the single laptop on even bigger data:
http://www.frankmcsherry.org/graph/scalability/cost/2015/02/04/COST2.html
For me, this set of publications is what finally took the hot air out of the big data balloon. Most people don't have big data and won't benefit (performance-wise,

not business-sexiness-wise) from using those systems. That's what I see when I read the links posted above.

The Truth: Where is Big Data Used Successfully

So is it completely useless to analyze large amounts of data? As I mentioned above - that depends on your goals. Here is an opposite opinion - a positive use-case for using big data:

http://www.ft.com/cms/s/0/304b983e-5a44-11e5-a28b-50226830d644.html
Charles Clover, China's Baidu searches for AI edge, Financial Times (September 14, 2015 4:24 am)

Here is the (probably copyrighted - sorry about that!) most important part in my opinion:

> The company has an advantage in deep-learning algorithms for speech recognition in that most video and audio in China is accompanied by text — nearly all news clips, television shows and films are close-captioned and almost all are available to Baidu and Iqiyi, its video affiliate.
>
> While a typical academic project uses 2,000 hours of audio data to train voice recognition, says Mr Ng, the troves of data available to China's version of Google mean he is able to use 100,000 hours.
>
> He declines to specify just how much the extra 98,000 hours improves the accuracy of his project, but insists it is vital.
>
> "A lot of people underestimate the difference between 95 per cent and 99 per cent accuracy. It's not an 'incremental' improvement of 4 per cent; it's the difference between using it occasionally versus using it all the time," he says.

So in this domain (speech recognition), using this algorithm (deep learning), big data of this high quality (closed captions) is successfully used. This also works well with image recognition done in neural networks over large corpora of tagged images. Google and Facebook have such datasets and they do it successfully. But not every company is Google or Facebook. Are you?

Conclusion

Before you get confused by big-data, please focus on the following points:

Big Data Systems are usually a combination of scalable file systems and scalable processing architectures. The household name in this field is Hadoop, which is based on these two Google papers:
1) http://static.googleusercontent.com/media/research.google.com/en//archive/gfs-sosp2003.pdf
2) http://static.googleusercontent.com/media/research.google.com/en//archive/mapreduce-osdi04.pdf
(although Google don't use these technologies nowadays:
3) http://cacm.acm.org/blogs/blog-cacm/177467-hadoop-at-a-crossroads/fulltext)

Most real data scientists admit they don't really use big data...
3) http://www.kdnuggets.com/2015/08/largest-dataset-analyzed-more-gigabytes-petabytes.html

... and new evidence shows that these systems are also not very efficient:
Paper by McSherry:
4) http://www.frankmcsherry.org/assets/COST.pdf
Two blog posts he made about this paper:
5) http://blog.acolyer.org/2015/06/05/scalability-but-at-what-cost/
6) http://www.frankmcsherry.org/graph/scalability/cost/2015/02/04/COST2.html

Big data is sometimes useful - here's an example. Understand whether you are in such a situation:
7) http://www.ft.com/cms/s/0/304b983e-5a44-11e5-a28b-50226830d644.html
If you are not - don't feel bad. You're in a good company :)

The Fine Print

I currently work for Dato, the creator of GraphLab Create. Our product is famous for analyzing massive datasets over a single laptop or a single machine. Nevertheless, these opinions do not represent Dato. Also, I got these opinions from experimenting with GraphLab Create, and seeing for myself that good data science can be done over less than petabytes, and in a single machine.