Big Data, Data Science and You—Demystifying some Big Ideas

Authors: Roy Pardee, JD MA[1]
Affiliations: [1] Group Health Research Institute

Introduction

The popular press is lately rife with references to 'Big Data' and 'Data Science'. We live on the cusp of a Revolution, we are told, where the ocean of data flowing out of ubiquitous and social computing will be harnessed to produce newfound Insights, to the profit of business and humankind generally.

Certainly the notion of using incidentally-produced data to yield valuable insights is one that we are used to—that describes quite a lot of our research work. But where do we fit into these trends? Have we been "Data Scientists" doing "Big Data" all along, and the rest of the world is only now catching up? Or are there really different and new things happening? More importantly—are there methods we should be appropriating to improve our work?

This poster defines three commonly encountered buzzphrases, relates them to familiar tools and technologies, and describes situations where we might want to employ them.

Big Data

What Is It?

BD is a particularly squishy term because it is negatively defined—data is 'big' or not in relation to available computing resources. BD tools/methods are where you go when you can't analyze your data in a timely fashion using available conventional resources & tools (say, a relational database like Oracle or a tool like SAS). Big data is "any data that is expensive to manage and difficult to extract value from."

Methods

The primary method at the bottom of the specialized tools employed for BD use is divide-and-conquer—that is, rather than trying to analyze an entire dataset in one go on a single server, it breaks the data and analysis up into N subsets, each of which is carried out on N servers. At the end of the subset analyses, the results are combined to describe the entire dataset. A more computer-sciencey way to describe this technique is *parallelization*. Because the subset analyses can be performed *at the same time*, on a large number of smaller servers, the subsets are said to be executed *in parallel*.

If these methods are so capable, why would anyone use anything else?

Because the programming interfaces are *very* bare-bones. Some tools store everything as text for example, forcing programmers to explicitly convert data items to numbers, dates, etc.

Are we currently using anything like it?

Yes—the Teradata database server that is the basis of Group Health's Enterprise Data Warehouse makes heavy use of parallelization. Data are split up onto 24 different 'amps' which act as independent servers, each of which execute queries in parallel. Teradata also supports a very rich programming interface—it essentially is a full relational database.

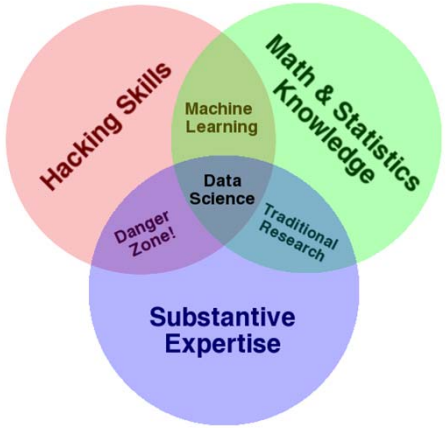
Data Science

What Is It?

The definition of Data Science is a matter of some controversy, but in general it entails three domains of knowledge/ability that come together in a single person:

- Programming skills—a Data Scientist is skilled at understanding and manipulating data to serve purposes other than those for which they were collected. They can integrate and clean data, and install configure and troubleshoot specialized software.
- Statistical/Analytical knowledge—A Data Scientist is competent with graphical, descriptive, inferential and machine learning methods for developing and testing explanatory models, and generally discriminating between signal and noise in datasets (of all sizes).
- Substantive expertise in the subject area of the data under analysis—medicine, banking, retail, telecommunications, etc.

One influential Data Scientist—Drew Conway put forth the following Venn diagram as a possible explanation for what Data Science is.



So Are We doing Data Science?

As an organization, GHRI is absolutely doing Data Science, and boasts several colleagues who have significant expertise in all three domains. In the larger Coop we actually have a Data Scientist employee title, which two of our colleagues in the Business Intelligence Competency Center group (nee Measurement & Analytics) currently hold.

But in the main I think we are pretty squarely in the 'Traditional Research' area between Stats and Substantive expertise. This should not be surprising—those domains are each huge, and many happy and productive careers have been spent in each one alone.

What's the relationship between BD and DS?

The tools and methods that a DS uses in her work need not be BD methods. A skilled DS would certainly not balk at using BD methods when necessary, but it's the data and the questions that drive the choice of methods.

Map/Reduce

What is it?

M/R is an algorithm for accomplishing the "divide-and-conquer" strategy mentioned above in the discussion of BD. It was proposed by two Google scientists back in 2004. The core of M/R is that you have a large number of independent computers, each doing one of two jobs. The Mappers take unstructured data as input and convert it into key/value pairs. For example they could read in files of book sales and use Author name as the Key, and the rest of the associated information as the Value.

Those KVPs are then routed to the Reducers according to their Keys. For example one computer may handle all authors whose last names start with 'A', another does the 'B' authors, and so on. The reducers then process the information in the Values (to e.g., count the number of sales for each author; or sum the amounts paid, etc.).

Once the Mappers have read all the files and the Reducers have done all their calculations, the reduced data is then collated to form the final answer.

What are some M/R Implementations?

M/R has been enormously successful for Google and many others, and has spawned an impressive number of open source implementations, including:

- Hadoop
- Pig
- Hive
- HBase
- Accumulo
- Riak
- Voldemort

These implementations frequently build on one another. For example, Pig is built on top of Hadoop, adding a data manipulation language (Pig Latin!) that makes Hadoop far easier to use. Hive is also built on top of Hadoop, adding a SQL-like language for data manipulation, which makes it easier for traditional database programmers to use.