10 things statistics taught us about big data analysis

Tags: Best Practices, Big Data, Jeff Leek, Overfitting, Statistics

---

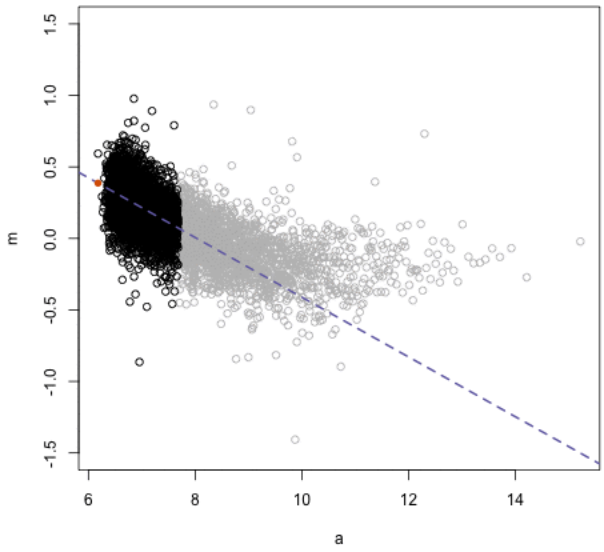
There are 10 ideas in applied statistics are relevant for big data analysis, focusing on prediction accuracy, interactive analysis and more.

comments

By Jeff Leek, (@jtleek)

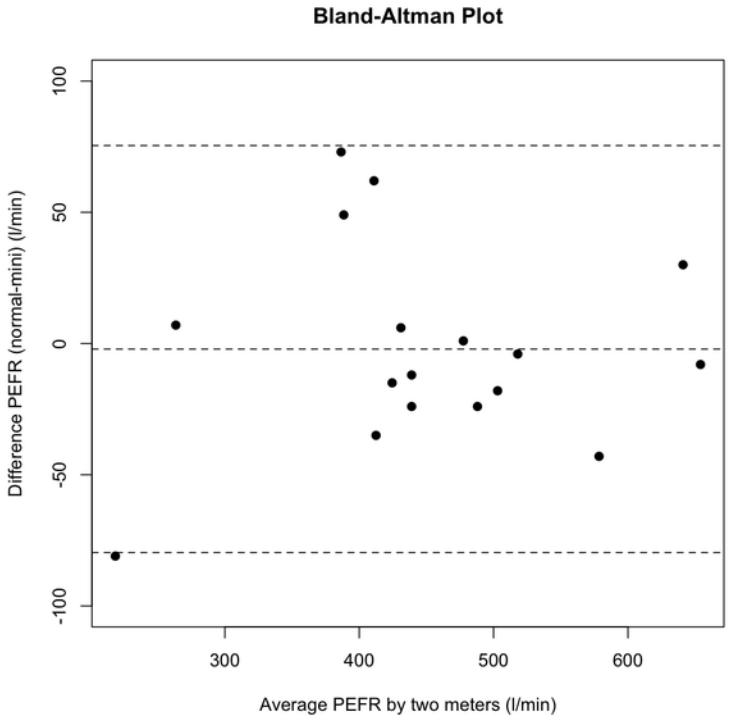
In my previous post I pointed out a major problem with big data is that applied statistics have been left out. But many cool ideas in applied statistics are really relevant for big data analysis. So I thought I'd try to answer the second question in my previous post: *"When thinking about the big data era, what are some statistical ideas we've already figured out?"* Because the internet loves top 10 lists I came up with 10, but there are more if people find this interesting. Obviously mileage may vary with these recommendations, but I think they are generally not a bad idea.

1. **If the goal is prediction accuracy, average many prediction models together**. In general, the prediction algorithms that most frequently win Kaggle competitions or the Netflix prize blend multiple models together. The idea is that by averaging (or majority voting) multiple good prediction algorithms you can reduce variability without giving up bias. One of the earliest descriptions of this idea was of a much simplified version based on bootstrapping samples and building multiple prediction functions - a process called bagging (short for bootstrap aggregating). Random forests, another incredibly successful prediction algorithm, is based on a similar idea with classification trees.
2. **When testing many hypotheses, correct for multiple testing**.
   This comic points out the problem with standard hypothesis testing when many tests are performed. Classic hypothesis tests are designed to call a set of data significant 5% of the time, even when the null is true (e.g. nothing is going on). One really common choice for correcting for multiple testing is to use the false discovery rate to control the rate at which things you call significant are false discoveries. People like this measure because you can think of it as the rate of noise among the signals you have discovered. Benjamini and Hochber gave the first definition of the false discovery rate and provided a procedure to control the FDR. There is also a really readable introduction to FDR by Storey and Tibshirani.
3. **When you have data measured over space, distance, or time, you should smooth**
   This is one of the oldest ideas in statistics (regression is a form of smoothing and Galton popularized that a while ago). I personally like locally weighted scatterplot smoothing a lot. This paper is a good one by Cleveland about **loess**. Here it is in a gif.



But people also like smoothing splines, Hidden Markov Models, moving averages and many other smoothing choices.

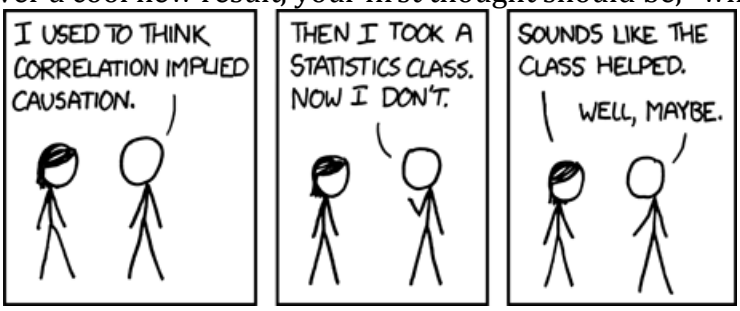
4. **Before you analyze your data with computers, be sure to plot it**
   A common mistake made by amateur analysts is to immediately jump to fitting models to big data sets with the fanciest computational tool. But you can miss pretty obvious things like this (Anscombe quartet) if you don't plot your data.



There are too many plots to talk about individually, but one example of an incredibly important plot is the Bland-Altman plot, (called an MA-plot in genomics) when comparing measurements from multiple technologies. R provides tons of graphics for a reason and ggplot2 makes them pretty.

5. **Interactive analysis is the best way to really figure out what is going on in a data set**
   This is related to the previous point; if you want to understand a data set you have to be able to play around with it and explore it. You need to make tables, make plots, identify quirks, outliers,

missing data patterns and problems with the data. To do this you need to interact with the data quickly. One way to do this is to analyze the whole data set at once using tools like Hive, Hadoop, or Pig. But an often easier, better, and more cost effective approach is to use random sampling . As Robert Gentleman put it "make big data as small as possible as quick as possible".

6. **Know what your real sample size is.**
   It can be easy to be tricked by the size of a data set. Imagine you have an image of a simple black circle on a white background stored as pixels. As the resolution increases the size of the data increases, but the amount of information may not (hence vector graphics). Similarly in genomics, the number of reads you measure (which is a main determinant of data size) is not the sample size, it is the number of individuals. In social networks, the number of people in the network may not be the sample size. If the network is very dense, the sample size might be much less. In general the bigger the sample size the better and sample size and data size aren't always tightly correlated.
7. **Unless you ran a randomized trial, potential confounders should keep you up at night**
   Confounding is maybe the most fundamental idea in statistical analysis. It is behind the spurious correlations like these and the reason why nutrition studies are so hard. It is very hard to hold people to a randomized diet and people who eat healthy diets might be different than people who don't in other important ways. In big data sets confounders might be technical variables about how the data were measured or they could be differences over time in Google search terms. Any time you discover a cool new result, your first thought should be, "what are the potential confounders?"
   

8. **Define a metric for success up front**
   Maybe the simplest idea, but one that is critical in statistics and decision theory. Sometimes your goal is to discover new relationships and that is great if you define that up front. One thing that applied statistics has taught us is that changing the criteria you are going for after the fact is really dangerous. So when you find a correlation, don't assume you can predict a new result or that you have discovered which way a causal arrow goes.
9. **Make your code and data available and have smart people check it**
   As several people pointed out about my last post, the Reinhart and Rogoff problem did not involve big data. But even in this small data example, there was a bug in the code used to analyze them. With big data and complex models this is even more important. Mozilla Science is doing interesting work on code review for data analysis in science. But in general if you just get a friend to look over your code it will catch a huge fraction of the problems you might have.
10. **Problem first not solution backward** One temptation in applied statistics is to take a tool you know well (regression) and use it to hit all the nails (epidemiology problems).

There is a similar temptation in big data to get fixated on a tool (hadoop, pig, hive, nosql databases, distributed computing, gpgpu, etc.) and ignore the problem of can we infer x relates to y or that x predicts y.

Original: simplystatistics.org/2014/05/22/10-things-statistics-taught-us-about-big-data-analysis/

Jeff Leek, is a professor at Johns Hopkins, where he does statistical research, writes data analysis software, curates and creates data sets, writes a blog about statistics, and work with amazing students who go do awesome things.