HSD5230 High Performance Computing
Midterm Exam

Miao Cai[*]

3/15/2019

1) What is the definition of big data? Although there are many possible answers here, I have communicated what I consider the right answer in class multiple times!

Apart from the three-V standards (volume, velocity, and variety), a more down-to-earth definition of big data is that the analytical tools or environment have to be changed or improved only because the size of the data is beyond the current the analytical tools or environment. For example:

- Data exceeds the memory I have on my PC
- Data exceeds the hard drive on my single server

2) What is the difference between inferential modeling approaches versus predictive modeling approaches? Come up with a clarifying example for each modeling approach (one inferential, one predictive).

Inferential modeling focuses on the inference and estimation about the parameters in the population. In comparison, predictive modeling aims at predicting the future and focuses on the accuracy of prediction.

Two clarifying examples on inferential and predictive modeling:

- Inferential: What are the significant factors associated with today's weather?
- Predictive: what is the weather like tomorrow based on the weather of the previous week?

3) Is machine learning superior to traditional approaches to statistical inference? When would you use machine learning versus traditional approaches to statistics? Give specific reasons to support your answer.

In my opinion, machine learning and traditional statistical inference are two streams of data science, and there are no superior in the current era.

The choice of machine learning or traditional statistics highly depends on the field and the types of question. In the field of public health, traditional 'data models' are viewed as the de facto way of statistics since researchers care more about odds ratio, intuition, and interpretability. By contrast, modern machine learning including consulting and recommending systems focus on the accuracy of prediction, therefore the 'algorithm modeling' is the de facto way of doing statistics.

4) Define the term 'data leakage' in the context of health care analytics (predictive models specifically). What are the symptoms of data leakage?

Data leakage is accidentally adding variables that are strongly associated with the outcome variable while these variables are impossible to obtain in real-life application (Wiens and Shenoy 2017).

[*]Department of Epidemiology and Biostatistics, College for Public Health and Social Justice, Saint Louis University. Email miao.cai@slu.edu

Below is some symptoms of data leakage:

- The predictive accuracy seems to be too high but it becomes very low when used in real-world situation,
- A strong association between a few of the predictors and the outcome variable

5) Explain 2 strengths of real world evidence compared with randomized clinical trials.

1. Large sample size: real world evidence (RWE) are often generated by administrative databases. The size of these databases cumulates and scales up as time goes by. Therefore, RWE is often large in sample size.
2. More external validity: since RWE has large sample size and covers a wider population, the results are more generalizeable to other population compared to RCTs.

6) Explain 2 strengths of randomized clinical trials compared to real world evidence.

1. Direct assignment of treatment: In contrast to self-selection often occurred in RWE (patient self-selection or physician selection), randomized clinical trials (RCTs) impose treatments on patients without any selection. In this way, RCT is free of patient's selection of treatment bias.
2. Randomisation: Randomization of treatment makes all other variables apart from the treatment cancel out. Therefore, RCTs are free of bias from unmeasured confounders.

7) The data.table package in R allows us to work on larger datasets than using base R or dplyr. Why/how does data.table allow us to work on larger data? Isn't data.table still limited to memory?

There are several ways of optimization in `data.table`:

- `data.table` uses references to the original object, instead of copying and replacing,
- `data.table` has index (keys) just like databases, which contributes to faster value accessing, group by and joins.

The `data.table` is still limited to memory, but it uses your RAM more efficiently.

8) What does is the difference between timing your code and profiling your code?

Timing code only records the total time of running all the code, but it does not tell you which part of the code is slow. In comparison, profiling code poinst out exactly the slow part (bottleneck) of your code.

9) When should we start profiling (or timing) our code?

We should start profiling code when the code takes a considerable long time to run and it will take less considerable time to profile it. We may also have to run similar code for several times since the analytical plan may be subject to minor changes.

10) Define 'embarrassingly parallel' calculations.

The 'embarrassingly parallel' calculation is when the action called in each iteration of the loop does not depend on any of the prior iterations of the loop. The order does not matter, and the user does not need to know the first result to calculate the second result.

11) Consider the following example: I have longitudinal medication data on 1 million patients in a healthcare system. I wish to calculate each patient's medication adherence for every year (a simple summary of how good each patient is at taking their medication). Is this an example of an 'embarrassingly parallel' calculation? Why or why not?

Yes, this is an 'embarrassingly parallel' calculation since the calculation on one patient does not influence the calculations on other patients.

12) Consider the following example: I have longitudinal medication adherence measures on 1 million patients in a healthcare system. I wish to fit a logistic regression to see if medication adherence predicts mortality among these patients. Is this an example of an 'embarrassingly parallel' calculation? Why or why not?

This is NOT an 'embarrassingly parallel' calculation since we need to pool the data together to get parameter estimates. In other words, we cannot fit logistic regression models for each patient and generalize that to other patients.

13) HDF5 data stores are one method to storing data on disk when using Pandas. Name 3 advantages HDF5 data stores have over structured text (like CSV) files when using Pandas:

- Multi-platform compatibility: HDF5 can be easily read and processed on R, Python, Matlab, C, Fortran, and others.
- Fast read and write speed
- Hierarchical structure: in contrast to the tables in relational databases, HDF5 organizes data in a hierarchical manner which is more natural.

14) What does 'lazy evaluation' mean? Why is that advantageous?

'Lazy evaluation' enables expressions to be evaluated until they are actually used. Lazy evaluation is advantageous because it can be more efficient programmatically: the expressions do not have to be implemented at the place where they were defined, so that they can be subject to changes according to other arguments.

15) Do the following frameworks generally use lazy evaluation in most of their functionality (yes or no)?
(a) Base R
(b) R: Data.table
(c) Python: Pandas
(d) Python: Dask

(a) Base R: Yes
(b) R: Data.table: Yes
(c) Python: Pandas: No
(d) Python: Dask: Yes

16) What is the smallest number of bits (not bytes!) you can use to represent the following numbers (presume you don't need negative numbers, these are unsigned)?
(a) 1
(b) 5
(c) 55
(d) 243
(e) 290
(f) 1025
(g) 67000

(a) 1: 8 bits
(b) 5: 8 bits
(c) 55: 8 bits
(d) 243: 8 bits
(e) 290: 16 bits
(f) 1025: 16 bits
(g) 67000: 32 bits

Correct answer:

a. `1 (1)`

b. `5 (3)`

c. `55 (6)`

d. `243 (8)`

e. `290 (9)`

f. `1025 (11)`

g. `67000 (17)`

17) What is the smallest number of bytes you can use to represent the following numbers (presume you don't need negative numbers, these are unsigned)?
(a) 15
(b) 265
(c) 67000
(d) 4294967298

(a) 15: one byte
(b) 265: 2 bytes
(c) 67000: 4 bytes
(d) 4294967298: 8 bytes

Correct answer:

a. `15 (1)`

b. `265 (2)`

c. `67000 (3)`

d. `4294967298 (5)`

18) Consider a dataset where I have hospital ID's that range from 1 to 200. If I am using Pandas and I wish to manually compress the data by downcasting the numeric data types, can I use int8 to represent this data? Why or why not?

No, it does not work since int8 has both positive and negative numbers and 1 to 200 is not within the range of -128 to 127.

19) Consider a dataset where I have hospital ID's that range from 1 to 200. If I am using Pandas and I wish to manually compress the data by downcasting the numeric data types, can I use uint8 to represent this data? Why or why not?

Yes, it works since uint8 can be up to 2*8 = 256 and 1-200 is within the range.

20) What are the two primary reasons Spark is faster than Hadoop?

- Spark operates in memory when it is possible (but can spill to disk),
- Spark reduces the number of read/write cycles compared to MapReduce without having to write to persistent storage.

21) Hadoop uses the Hadoop distributed file system (HDFS) to solve the distributed data problem, and uses MapReduce to solve the distributed analytic problem. Which of these two aspects of Hadoop does Spark replace/replicate?

Spark succeeded the popular Hadoop MapReduce computation framework and can be viewed as a 2nd-generation MapReduce.

22) Define vertical versus horizontal resource scaling in high performance computing.

- Vertical resource scaling: adding more RAM, hard drive space, or buying more advanced processors on a single computer or server,
- Horizontal resource scaling: using more computers, distributed computation systems, or parallelization.

23) Name 3 differences between SQL and NoSQL approaches to data storage.

1. Scalability: non-SQL database is more horizontally scalable by using Map-Reduce while SQL approaches are more vertically scalable,
2. Detailed database model: nonSQL approaches do not need a well-defined database model (a dynamic schema). In contrast, SQL approaches require detailed database design (static or predefined schema), which requires a long development time,
3. Cost: nonSQL approaches are generally open source and have low costs. Instead, SQL approaches typically requires a long development time and higher financial input.

References

Wiens, Jenna, and Erica S Shenoy. 2017. "Machine Learning for Healthcare: On the Verge of a Major Shift in Healthcare Epidemiology." *Clinical Infectious Diseases* 66 (1): 149–53.