Stat243: Problem Set 6, Due Friday November 6

October 30, 2020

This covers Units 8 and 9.
It's due **as PDF submitted to Gradescope** and submitted via GitHub at 10 am on Nov. 6.
Comments:

1. The formatting requirements are the same as previous problem sets.

2. Please note my comments in the syllabus about when to ask for help and about working together. In particular, **please give the names of any other students that you worked with on the problem set and indicate in comments any ideas or code you borrowed from another student.**

Problems

1. This is the October 30 Section work on comparing R and Python. Please submit your answers at the end of section via the Gradescope group problem specific to the section work. Do not submit your work here.

2. Consider the full Wikipedia traffic data for October-December 2008 (available in *var/local/s243/wikistats/dated_2017* on any of the low (default) partition SCF cluster nodes).

   (a) Explore the variation over time in the number of visits to Barack Obama-related Wikipedia sites, based on searching for "Barack_Obama" on English language Wikipedia pages. You should use Dask to do the reading and filtering. Then group by day-hour (it's fine to do the grouping/counting in Python in a way that doesn't use Dask data structures). You can do this either in an interactive session using *srun* or a batch job using *sbatch*. And if you use *srun*, you can run Python itself either interactively or as a background job. Time how long it takes to read the data and do the filtering to get a sense for how much time is involved working with this much data. Once you have done the filtering and gotten the counts for each day-hour, you can simply use standard R or Python code on your laptop to do some plotting to show how the traffic varied over the days of the full October-December time period and particularly over the hours of November 3-5, 2008 (election day was November 4 and Obama's victory was declared at 11 pm Eastern time on November 4).

   (b) Extra credit: Carry out some analyses of how the traffic varies by language or do some other in-depth analysis of the Wikipedia data (it doesn't have to involve Barack Obama), addressing a question of interest to you.

   Notes:

   - Note that I'm not expecting you to know any more Python than we covered in the Unit 7/8 material on Dask and in Section, so feel free to ask for help (and for those of you who know Python to help out) on Python syntax on Piazza or in office hours.

- There are various ways to do this using Dask bags or Dask data frames, but I think the easiest in terms of using code that you've seen in Unit 8 is to read the data in and do the filtering using a Dask bag and then convert the Dask bag to a Dask dataframe to do the grouping and summarization. Alternatively you should be able to use *foldby()* from *dask.bag*, but figuring out what arguments to pass to *foldby()* is a bit involved.
- Make sure to test your code on a portion of the data before doing computation on the full dataset. **Reading and filtering the whole dataset will take something like 60 minutes with 16 cores. You MUST test on a small number of files on your laptop or on one of the stand-alone SCF machines (e.g., radagast, gandalf, arwen) before trying to run the code on the full 120 GB (zipped) of data.** For testing, the files are also available in */scratch/users/paciorek/wikistats/dated_2017*.
- When doing the full computation via your SLURM job submission:
  - You must read the data from */var/local/s243/wikistats/dated_2017* (which is available on all the machines on the SCF cluster, so you don't need to copy it, unlike in PS5).
  - Please do not use more than 16 cores in your SLURM job submissions so that cores are available for your classmates. If your job is stuck in the queue you may want to run it with 8 rather than 16 cores.
  - As discussed, when you use *sbatch* to submit a job to the SCF cluster or *srun* to run interactively, you should be using the *–cpus-per-task* flag to specify the number of cores that your computation will use. In your Python code, you can then either hard-code that same number of cores as the number of workers or (better) you can use the SLURM_CPUS_PER_TASK UNIX environment variable to tell Dask how many workers to start.
  - Note that the SCF machines will be down 7-10 am on Friday Oct. 30.

3. Using the Stack Overflow database (http://www.stat.berkeley.edu/share/paciorek/stackoverflow-2016.db), try our SQL question from class about determining the oldest users. Time how long the two approaches (sorting versus choosing values greater than some cutoff) are. Then create an index and run the timing again. The Stack Overflow SQLite database is ~ 650 MB on disk, which should be manageable on most of your laptops, but if you run into problems, you can use an SCF machine.

4. Using the Stack Overflow database, write SQL code that will determine which users have asked html-related questions but not css-related questions. Those of you with more experience with SQL might do this in a single query, but it's perfectly fine to create one or more views and then use those views to get the result as a subsequent query. Report how many unique such users there are. There are various ways to do this, of which we've only covered some approaches in the class material.

5. The goal of this problem is to think carefully about the design and interpretation of simulation studies, which we'll talk about in Unit 9, in particular in Section on Friday November 6. In particular, we'll work with Cao et al. (2015), an article in the Journal of the Royal Statistical Society, Series B, which is a leading statistics journal. The article is available as *cao_etal_2015.pdf* under the *ps* directory on Github. Read Section 1, Section 2.1, and Section 4 of the article. Also read Sections 2.1-2.2 of Unit 9

   You don't need to understand their method for fitting the regression [i.e., you can treat it as some black box algorithm] or the theoretical development. In particular, you don't need to know what an estimating equation is - you can think of it as an alternative to maximum likelihood or to least squares for estimating the parameters of the statistical model. Equation 3 on page 759 is analogous to taking the sum of squares for a regression model and differentiating with respect to β. To find $\hat{\beta}$ one sets the equation equal to zero and solves for β. As far as the kernel, its role is to weight each pair of observation and covariate value. This downweights pairs where the covariate is measured at a very different time than the observation.

Briefly (a few sentences for each of the three questions below) answer the following questions.

(a) What are the goals of their simulation study and what are the metrics that they consider in assessing their method?

(b) What choices did the authors have to make in designing their simulation study? What are the key aspects of the data generating mechanism that might affect their assessment of their method?

(c) Consider their Tables 1 and 3 reporting the simulation results. For a method to be a good method, what would one want to see numerically in these columns?

In Section on November 6, we'll talk in more detail about this simulation study.