Data Extraction Evaluation

Matthew Leonawicz

December 24, 2014

0.0.1 Results

All points? No point.

Using the sample mean is helpful as a data reduction strategy while not being harmful in terms of representativeness. The possible "tradeoff" itself appears to be largely a false dichotomy. There is no benefit to computing the mean of all pixels in the example map layer.

How many samples do we really need?

```
## Error in eval(expr, envir, enclos):  object 'no.knit' not found
## Error in ggplot(p, aes(x = Percent_Sample, y = Pval, group = Type, colour = Type)):  object
'p' not found
## Error in eval(expr, envir, enclos):  object 'g' not found
## Error in eval(expr, envir, enclos):  object 'g' not found
## Error in print(g):  object 'g' not found
## Error in eval(expr, envir, enclos):  object 'no.knit' not found
```

In this example even a two percent subsample of the original non-NA data cells is small enough to limit us to a five percent probability of obtaining a mean that differs from the mean computed on the full dataset by an amount equal to or greater than the smallest discrete increment possible (0.1 degrees Celsius for SNAP temperature data) based simply on the number of significant figures present. Furthermore, even for nominal sample sizes, the 0.05 probability is one almost strictly of minimal deviation (0.1 degrees). The probability that a sample mean computed on a subsample of the map layer deviates enough from the population mean to cause it to be rounded to two discrete incremental units from the population mean (0.2 degrees) is essentially zero (except if using crudely small sample sizes).

Although a two percent subsample appears sufficient for this criterion, lets use a five percent subsample for illustration. This is clearly overkill in this example since the p-value attenuates to the range of 0.019 to 0.029 by around 2.5 percent subsampling.

How much faster does this make things go?

Compute time for the mean is of course affected by the sample size.

```
## Error in eval(expr, envir, enclos):  object 'd.sub' not found
## Error in eval(expr, envir, enclos):  object 'd.sub' not found
## Error in eval(expr, envir, enclos):  object 'd.sub' not found
## Error in eval(expr, envir, enclos):  object 'd.sub' not found
```

```
## Error in microbenchmark(sum(s005pct)/length(s005pct), sum(s010pct)/length(s010pct), :  object
's100pct' not found
## Error in eval(expr, envir, enclos):  object 'mb3' not found
## Error in eval(expr, envir, enclos):  object 'no.knit' not found
## Error in autoplot(mb3):  object 'mb3' not found
## Error in eval(expr, envir, enclos):  object 'no.knit' not found
```

Using optimal subsampling to estimate the mean achieves speed improvements orders of magnitude greater than what can be achieved through strictly algorithmic changes to how the mean is computed on the full dataset, though those help immensely as well, also by many orders of magnitude. Sampling is vastly more effective, but both approaches can be combined for maximum benefit.

```
## Error in mean(s005pct):  object 's005pct' not found
## Error in eval(expr, envir, enclos):  object 'mb4' not found
## Error in print(mb4):  object 'mb4' not found
## Error in print(mb4):  object 'mb4' not found
## Error in eval(expr, envir, enclos):  object 'med' not found
## Error in eval(expr, envir, enclos):  object 'no.knit' not found
## Error in autoplot(mb4):  object 'mb4' not found
## Error in eval(expr, envir, enclos):  object 'no.knit' not found
```

Similar to above, below are the median compute times for the mean using (1) the full data while removing NAs, (2) the sum divided by the length after NAs removed, (3) the mean of a subsample, and (4) a combination of (2) and (3).

```
## Error in eval(expr, envir, enclos):  object 'no.knit' not found
## Error in data.frame(x = names(med), y = med):  object 'med' not found
## Error in eval(expr, envir, enclos):  object 'no.knit' not found
```

Here is the same plot after removing the first bar to better show the relative compute time for the other three methods.

```
## Error in eval(expr, envir, enclos):  object 'no.knit' not found
## Error in data.frame(x = names(med)[-4], y = med[-4]):  object 'med' not found
## Error in eval(expr, envir, enclos):  object 'no.knit' not found
```

How does the benefit extend to extractions on maps at different extents, data heterogeneity, climate variables, or for other common statistics such as the standard deviation? These are open questions at the moment, but for one thing, I expect more samples are needed for precipitation than temperature. I also expect more samples needed to estimate parameters with higher moments.