Assessed Coursework 1

CID 01252821

Question 1

In the paper “Statistical paradises and paradoxes in big data (I): Law of large populations, big data paradox, and the 2016 US presidential election” written by Xiao-Li Meng (2018) and part of the The Annals of Applied Statistics 12(2), 685-726, the author addresses the question “Which one should I trust more: a 1% survey with 60% response rate or a self-reported administrative dataset covering 80% of the population?”. He suggests a framework to answer this question by developing measures for data quality, the Law of Large Populations (LLP) is introduced, and the Big Data Paradox: the more the data, the surer we fool ourselves, is discussed.

The author starts with pointing out that more data didn’t turn out to be the statistical paradise many expected, mainly due to complications with handling the volume easily on personal computers, the variety that challenges the models, including the most sophisticated ones, and the velocity needed to handle such problems. He points out that Quality control is important and emphasizes the purpose of this paper is to focus on population inferences from Big Data. To elaborate: to shift from our traditional focus on assessing probabilistic uncertainty (standard error) to the practice of ascertaining systematic error in non-probabilistic Big Data (relative bias). This shift is later prooved to be needed via theoretical and empirical evidence.

Meng continues with suggesting a fundamental identity for data quality-quantity tradeoff for using sample averages to estimate population averages, attempting to answer another important question: “How to compare two datasets with different quantities and different qualities?”. This identity is expressed as a product of a data quality measure, a data quantity measure, and a problem difficulty measure. These and only these three factors determine the estimation error, with data quality being the most critical one. After that, the author defines a data defect index (d.d.i.), where the smaller the data defect index, the higher the data quality is.

Then Meng introduces the concept of the Law of Large Populations (LLP) stating that as the sample size increases and control is lost over probabilistic sampling, the population size becomes more accurate than the sample size in estimating the error. A Big Data Paradox is outlined: “The bigger the data, the surer we fool ourselves.”, in relation to the “warning label” that the assertion that “the population size is not relevant for inference concerning population means and alike, as long as N is sufficiently large” is only valid if there is strict control of the sampling scheme.

We can see these concepts and results applied to binary outcomes and to the 2016 US presidential elections (pre-election over-confidence). An in depth analysis is carried out showcasing the bias in reporting propensity, using the population of eligible voters and not of the actual ones. A few formulas are introduced to show how non-response biases affect the effective sample size.

Finally, the Euler’s identity is observed, alongside its applications in Monte Carlo and Quasi Monte Carlo methods. The idea of possibly reducing d.d.i. while enhancing data privacy is also showcased in relation to the identity, as well as the concept of synthetic datasets is suggested as a way of preserving confidentiality. In the end, Meng defines the problem of Indiviualized statistics which is described as an ultimate challenge for Statistics, or “How to build a meaningful theoretical foundation for inference and prediction without any direct data?”.

Overall, the main objective of the paper is to explore the challenges and implications that are faced when working with Big Data, and to suggest a framework for addressing questions about data quality, in help of identifying the differences between sample averages and population averages in the context of Big Data.

Question 2

Part a

From the definition of conditional probability we have:

$$\phi_1 = P(R_j = 1 | X_j = 1) = \frac{P(R_j = 1 \cap X_j = 1)}{P(X_j = 1)} = \frac{P(R_j = 1 \cap X_j = 1)}{p_X},$$

$$\phi_0 = P(R_j = 1 | X_j = 0) = \frac{P(R_j = 1 \cap X_j = 0)}{P(X_j = 0)} = \frac{P(R_j = 1 \cap X_j = 0)}{1 - P(X_j = 1)} = \frac{P(R_j = 1 \cap X_j = 0)}{1 - p_X}$$

Let's find expressions for ψ_1 and ψ_0. From Bayes Theorem we have:

$$\psi_1 = P(X_j = 1 | R_j = 1) = \frac{P(R_j = 1 | X_j = 1) P(X_j = 1)}{P(R_j = 1)} = P(R_j = 1 | X_j = 1) \frac{p_X}{f} = \phi_1 \frac{p_X}{f},$$

$$\psi_0 = P(X_j = 0 | R_j = 1) = \frac{P(R_j = 1 | X_j = 0) P(X_j = 0)}{P(R_j = 1)} = P(R_j = 1 | X_j = 0) \frac{1 - p_X}{f} = \phi_0 \frac{1 - p_X}{f}$$

Now, for f we also have:

$$f = P(R_j = 1) = P(R_j = 1 \cap X_j = 0) + P(R_j = 1 \cap X_j = 1) = (1 - p_X)\phi_0 + p_X \phi_1$$

We get the following system for ϕ_0 and ϕ_1:

$$\begin{cases} f = (1 - p_X)\phi_0 + p_X \phi_1 \\ \Delta_R = \phi_1 - \phi_0 \end{cases} . \tag{1}$$

Substituting the given values $p_X = 0.51$, $f = 0.1$, $\Delta_R = -0.01$:

$$\begin{cases} 0.1 = 0.49\phi_0 + 0.51\phi_1 \\ -0.01 = \phi_1 - \phi_0 \end{cases} . \tag{2}$$

or we get $\phi_1 = 0.0951$ and $\phi_0 = 0.1051$.

Part b

If we assume the file `calisota_votes.csv` on the Athena local file system is located at my user directory `/users/jtl22`, then in order to upload it to the `/shared_data` folder on HDFS (assuming the correct location is the current location of the file, or the `/shared_data` folder on HDFS), we would use the following command:

```
hadoop fs -put /users/jtl22/calisota_votes.csv /shared_data
```

Part c

The file `mapper.py` is:

```
#!/usr/bin/env python

import sys

for line in sys.stdin:
    # remove leading or trailing whitespace
    line = line.strip()
    # extract input values
    x, r, d = line.split(",")
    # key-value pair for each district <d, x_r>
    print(d, f"{int(x)}_{int(r)}", sep="\t")
```

Producing key-value pairs such as `<d, <x, r>>`.

The file `reducer.py` is:

```
#!/usr/bin/env python

import sys
from collections import Counter


print("District", "pX", "f", "phi1", "phi0", "DR", "psi1", "neff", sep="\t")

current_key = None
key = None
current_count = Counter()


def calculate_estimates(current_count, current_key):
    p = current_count["x1"] / current_count[current_key]
    f = current_count["r1"] / current_count[current_key]
    f1 = current_count["r1_x1"] / current_count["x1"]
    f0 = (current_count["r1"] - current_count["r1_x1"]) / (
        current_count[current_key] - current_count["x1"]
    )

    dr = f1 - f0
    psi1 = f1 * p / f
    neff = ((f / dr) ** 2) / (p * (1 - p))

    return p, f, f1, f0, dr, psi1, neff


for line in sys.stdin:
    line = line.strip()
    key, value = line.split("\t")
    x, r = map(int, value.split("_"))

    if current_key == key:
        current_count[key] += 1
        current_count["x1"] += x
        current_count["r1"] += r
        current_count["r1_x1"] += x * r
    else:
        if current_key is not None:
            p, f, f1, f0, dr, psi1, neff = calculate_estimates(
                current_count, current_key
            )
            print(current_key, p, f, f1, f0, dr, psi1, neff, sep="\t")

        current_count = Counter()

        current_count[key] += 1
        current_count["x1"] += x
        current_count["r1"] += r
        current_count["r1_x1"] += x * r

        current_key = key


if current_key == key:
    p, f, f1, f0, dr, psi1, neff = calculate_estimates(current_count, current_key)
    print(current_key, p, f, f1, f0, dr, psi1, neff, sep="\t")
```

Shell file to execute the Hadoop job:

```
#!/bin/bash

hadoop jar $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-3.3.4.jar \
  -libjars $HADOOP_HOME/share/hadoop/tools/lib/hadoop-streaming-3.3.4.jar \
  -input /shared_data/calisota_votes.csv \
  -output /users/jtl22/output \
  -file mapper.py \
  -mapper 'python3 mapper.py' \
  -file reducer.py \
  -reducer 'python3 reducer.py
```

After executing the job we get the following estimates:

District	p_X	f	ϕ_1	ϕ_0	Δ_R	ψ	n^*_{eff}
G	0.5503	0.0305	0.0349	0.0252	0.0098	0.6295	39
M	0.4496	0.0387	0.0251	0.0499	-0.0248	0.2910	10
SC	0.5001	0.0101	0.0100	0.0102	-0.0002	0.4956	12130

Part d

District Goosetown:

- eligible voters: 4629451
- $p_X = 0.55$ - the population vote share for Donald Duck

- $f = 0.03 = 3\%$ - opinions from up to 3% of the eligible voter population or n ~ 138884
- $\Delta_R = 0.01$ - reporting bias in absolute terms
- $n^*_{eff} = 39$ - effective sample size. This represents an over 99.99% loss of sample size compared to n = 4629451, resulting in corresponding margin of error $M_e = 16\%$ which is 400 times larger than $\frac{1}{\sqrt{n}} = 0.04\%$ (the margin of error from using the apparent size n = 4629451). This would lead to gross overconfidence in what the data would actually tell us.

District Mousetown:

- eligible voters: 3599184
- $p_X = 0.45$ - the population vote share for Donald Duck
- $f = 0.04 = 4\%$ - opinions from up to 4% of the eligible voter population or n ~ 143967, percentage of people who honestly report their plans
- $\Delta_R = -0.02$ - reporting bias in absolute terms
- $n^*_{eff} = 10$ - effective sample size. This represents an over 99.99% loss of sample size compared to n = 3599184, resulting in corresponding margin of error $M_e = 32\%$ which is 640 times larger than $\frac{1}{\sqrt{n}} = 0.05\%$ (the margin of error from using the apparent size n = 3599184). This would lead to gross overconfidence in what the data would actually tell us.

District St. Canard:

- eligible voters: 2057818
- $p_X = 0.5$ - the population vote share for Donald Duck
- $f = 0.01 = 1\%$ - opinions from up to 1% of the eligible voter population or n ~ 20578, percentage of people who honestly report their plans
- $\Delta_R = -0.0002 \approx 0$ - reporting bias in absolute terms
- $n^*_{eff} = 12130$ - effective sample size. This represents over 99.41% loss of sample size compared to n = 2057818, resulting in corresponding margin of error $M_e = 0.9\%$ which is ~13 times larger than $\frac{1}{\sqrt{n}} = 0.07\%$ (the margin of error from using the apparent size n = 2057818).

Conclusion:

We can notice that for district Goosetown there is a slight bias towards voting for Donal Duck (a goose), with population vote share for Donald Duck 0.55. It is also shown that there is gross overconfidence in the results.

Analogically, for Mousetown there is a slight bias towards voting for Miceky Mouse (a mouse), with population vote share for Mickey Mouse 0.55. Again, it is shown there is gross overconfidence in the results.

For St.Canard, the district with the smallest loss of sample size, reports an unbiased result of 0.5 population vote share for Donald Duck and Mickey Mouse as well. As expected, the margin of error M_e is only 13 times larger than the margin of error from using the apparent size of the district eligible voters. Compared to the two other districts with differences of 400 and 640 times, this district has a significantly smaller overconfidence. This district seems to show unbiased results towards Donald Duck or Mickey Mouse, but at the same time has less number of eligible voters.

If the newspaper is considered moderate, it would suggest there would be no strong preferences towards Mickey or Donald which could bias the readers, as well as the newspaper wouldn't attract readers based on strong preferences and the readers would be considered neutral. The St. Canard district seems to represent a moderate collection of voters balancing each other, while the other two districts obviously have preferences towards their "leaders", which might be explained by the fact that Donald Duck is a goose and Goosetown is probably biased by this (that's confirmed by our results), and the same goes for Mickey Mouse who is a mouse and Mousetown is probably biased by this. This leads me to think that the fact that CPI only contacted readers of the "Calisota Post" doesn't explain the difference in the results between the CPI poll and the election, but rather the results obtained for the biased districts Goosetown and Mousetown.