Total Error: Using ML to Measure Total Exposure*

Gaurav Sood†

Abstract

The consequences of measurement error are well studied. One unusual case that is less studied is when the measure of interest is at a different level than at which a measurement is made. For instance, in browsing data, the measurement is taken at the domain level but the measure of interest is at the user level, e.g., total time spent on pornographic websites. Small biases in measurement at the domain level can dramatically affect the validity of the user-level measures. We illustrate the problem through a small simulation and formalize and summarize the issue in this small note.

*Simulation script can be downloaded from http://github.com/soodoku/total_error.

†Gaurav can be reached at gsood07@gmail.com

Say that we want to measure how much time people spend consuming pornography online (e.g., Shen and Sood 2023). To measure that, say that we use a model to predict whether or not a domain hosts pornography (e.g., Chintalapati and Sood 2022). Here below, we discuss some of the concerns about using scores from such a model and discuss ways to address the issues.

Let's say we have n users. We use i to iterate over them. Let k denote the total number of unique domains visited by all the users during the observation window. And we use j to iterate over the domains. Let's denote the number of visits to domain j by user i by $c_{ij} = 0, 1, 2, \ldots$. And let's denote the total number of unique domains a person visits ($\sum (c_{ij} == 1)$) using t_i. Lastly, let's denote predicted labels about whether or not each domain hosts pornography by p, so we have $p_1, \ldots, p_j$.

Say there are five domains with predicted labels $p : 1_1, 1_2, 1_3, 1_4, 1_5$. Let's assume that for the chosen classification threshold, the False Positive Rate (FPR) is 10% and the False Negative Rate (FNR) is 7%. Let's say user one visits the first three sites once and user two visits all five sites once. Given 10% of the predictions are false positives, the total measurement error in user one's score $= 3 * .10$ and the total measurement error in user two's score $= 5 * .10$. More generally, the total number of false positives increases as a function of predicted $1s$. But say that some domains have a predicted label of 0. The error incurred on those domains is a false negative. And the total number of false negatives increases with predicted $0s$. Combining the two points, the bias for user i is:

$$\sum_{1}^{k} c_{ij} * (p_j == 1) * (FPR) - c_{ij} * (p_j == 0) * (FNR)$$

Formalizing allows us to clearly see that the net bias is a function of $FPR - FNR$ and c_{ij}. Keeping $FPR - FNR$ constant, the bias grows in c_{ij}. When c_{ij} is right-skewed as is common in browsing data, misclassifying domains that people visit a lot can be very

expensive—it may even change inferences wholesale.

To illustrate the problem, we conduct a small simulation. We simulate measure for a 1000 respondents. We start by randomly choosing the number of domains visited by a respondent. We randomly sample a number between 5 and 1000. Second, we simulate a multivariate normal with a covariance of .9 to create two columns and take one of the columns to reflect the true measure. To the $\hat{y}$ column, we add a small bias— .1 of the standard deviation of the true measure. We then aggregate this data at the respondent level and calculate the sum. As expected, the correlation between the means of the true and predicted measures in the aggregated data is also unaffected. But the correlation between the sums (which can be seen as a tally of total visits) is dramatically lower.

One way to tackle the issue is to use different probability cutoffs for classification. Different probability cutoffs generate different FNR and FPR rates and allow us a way to provide bounds for the inferences.

To directly tackle the problem of skew, we could tweak the cost function of the domain-level model such that the cost of each error is proportional to usage. But given the skew, it would put a metric ton of weight on the features of too few domains. And that may mean that the performance of the model is pretty bad. A better, simpler solution may be to hand code commonly visited domains.

References

Chintalapati, Rajashekar and Gaurav Sood. 2022. "piedomains: Predict the kind of content hosted by a domain based on domain name and content.".
URL: *https://github.com/themains/piedomains/*

Shen, Lucas and Gaurav Sood. 2023. "Holier Than Thou: Partisan Gap in the Consumption of Pornography Online.".