Secure MPC Protocol for Standard Deviation

February 6, 2019

1 Introduction

Recall that for a database of size n with data points x_i indexed by $i \in 1, \ldots, n$, the sample variance σ^2 is defined to be

$$\sigma^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2,$$

where $\bar{x}$ is the sample mean:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i.$$

The sample standard deviation of a database, σ, is the square root of the sample variance.

In the setting of MPC, rather than considering a single database of entries, we consider the case in which there is a set of parties where each party P_i holds entry x_i. The parties wish to compute the standard deviation of their inputs, but do not wish to reveal their inputs to anyone. Our protocol relies on the composition of basic arithmetic operations which are secure and relies on the modular composition properties of semi-honest MPC protocols.[1]

2 Secure Building Blocks

The following building blocks that are implemented in JIFF are useful to us here:

- share(input): secret-shares input between all parties
- share1.sadd(share2): secure addition of shares, which returns a share of results.
- share1.ssub(share2): secure subtraction of shares, which returns a share of results.
- share1.smult(share2): secure multiplication of shares, which returns a share of results.
- share.cmult(n): secure multiplication of share by a constant

[1] I should probably find those and make sure I actually cite a thing that says these are real...

3 Basic Implementation

The most naive implementation of the standard deviation would be to just directly implement the standard formula listed in the introduction using our secure building blocks. However, since secure multiplication or taking the secure square root of secret shares is much more computationally expensive than, say, secure addition, it is useful to try to optimize the calculations. First note that it sufficient to calculate the unweighted variance,

$$\tilde{\sigma}^2 = \sum_{i=1}^{n} (x_i - \bar{x})^2,$$

in a secure manner, and then use post-processing in the clear to determine the standard deviation from there. I.e., release $\tilde{\sigma}^2$ publicly and have all parties individually compute

$$\sigma = \sqrt{\frac{\tilde{\sigma}^2}{N}}.$$

Furthermore, since the subtraction of $\bar{x}$ from x_i in the calculation of the residual may result in negative numbers, which are more computationally intensive to deal with, we can remove the use of negative numbers entirely (except in the subtraction operation itself) by expanding $\tilde{\sigma}^2$ to

$$\tilde{\sigma}^2 = \sum_{i=1}^{n} (x_i^2 - 2x_i\bar{x} + \bar{x}^2).$$

Note also that

$$\begin{aligned} \sum_{i=1}^{n} 2x_i\bar{x} &= 2\bar{x}\sum_{i=1}^{n} x_i \\ &= 2\left(\frac{\sum_{i=1}^{n} x_i}{n}\right)\sum_{i=1}^{n} x_i \\ &= \frac{2}{n}\left(\sum_{i=1}^{n} x_i\right)^2. \end{aligned}$$

Then, setting $s = \sum_{i=1}^{n} x_i$ and $t = \sum_{i=1}^{n} x_i^2$, we can calculate

$$
\begin{aligned}
\tilde{\sigma}^2 &= \sum_{i=1}^{n} (x_i^2 - 2x_i\bar{x} + \bar{x}^2) \\
&= \sum_{i=1}^{n} x_i^2 - 2\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)\sum_{i=1}^{n} x_i + \sum_{i=1}^{n}\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right)^2 \\
&= t - \frac{2}{n}s^2 + \frac{1}{n^2}\sum_{i=1}^{n} s^2 \\
&= t - \frac{1}{n}s^2
\end{aligned}
$$

in a private manner. We then use a public post-processing step to calculate σ from $\tilde{\sigma}^2$. Note that this is in essence just calculating standard deviation using the alternative formulation that

$$\sigma^2 = \mathrm{E}[X^2] - (\mathrm{E}[X])^2.$$