eQTL project

Maurice Lubetzki
Yongjin Park
Secure AI Labs

1. Study explanations

The purpose of this study is to test differential privacy on a computer biology use case. Yongjin works on a project where SNPs data are collected and linked to genes expressions supposed to drive Alzheimer disease.

A simple linear regression is used to predict gene expressions based on SNPs samples. Thus, we implement a differentially private simple linear regression and compare the results to non-private one. A lot of studies are still on their way to compute a differentially private linear regression and writing from scratch a code for differentially private linear regression is computationally very expensive. Therefore, I opted to use a tensorflow library, *privacy*, to conduct the experiment.

The first part of the project consists in simulating the SNPs data and gene expression and understand how a simple linear regression with Differential privacy behave. To do so, we sample a random matrix of 0,1 or 2 elements and intentionally correlate it to gene expressions. The math behind this is elaborated in the code, via the *DataLoader* Function.

The differential privacy optimizer we use is set with a clip gradient and a noise multiplier parameter. The former is used to avoid gradient divergence that could happen due to the noise added at each iteration and the latter is the amount of Gaussian noise added to the gradient.

The amount of privacy budget achieved after each experiment remains constant, regardless the size of the data set. This is because it only depends on the noise multiplier and the number of iterations which are constant in our experiment. The privacy budget might be very important although a "large value […] could still mean good practical privacy"[1].

To ensure that our model is correct we have fitted multiple linear regressions using *sklearn* and compared the R^2s of both the latter library and *tensorflow*'s.

[1] http://www.cleverhans.io/privacy/2019/03/26/machine-learning-with-differential-privacy-in-tensorflow.html

First, we set relevant parameters; number of iterations and learning rate that leads to a robust linear regression. In this case, we used respectively, 700 and 0.025. Then, we implemented the differentially private linear regression with the same parameters and we checked if we could keep the same robustness.

The next step of the study will be to implement our algorithm to real data sets to validate the conclusions we had on the simulated one.

2. Results and interpretation

DP = Differential privacy

	N	P	R_2	R_2_DP	R_2_adj	R_2_adj_DP	epsilon
0	100	10	-0.0646056	-0.0485168	-2.48539	-2.49176	1678.16
1	10000	10	-0.267697	-0.349098	-0.959825	-0.853831	1678.16
2	100	10	-0.295457	-0.275067	-1.73812	-1.77301	1678.16
3	1000	10	0.0458412	0.0495668	-0.0504196	-0.0500454	1678.16
4	9000	50	0.18425	0.163755	0.00601131	-0.00132726	1678.16
5	8000	50	0.148088	0.139863	-0.00882683	-0.0112698	1678.16
6	1000	50	0.196575	0.201879	-0.265586	-0.262803	1678.16
7	6000	50	0.263374	0.235547	0.0287979	0.0143091	1678.16
8	2000	50	0.287974	0.255239	-0.0400029	-0.0601687	1678.16
9	7000	50	0.180181	0.145559	-0.00284624	-0.0145357	1678.16
10	3000	50	0.0557282	0.00457317	-0.0890288	-0.0923986	1678.16
11	5000	50	0.19679	0.154595	-0.0142074	-0.0298504	1678.16
12	4000	50	0.129962	0.110302	-0.0470312	-0.0520619	1678.16
13	2000	100	0.999979	0.937276	0.999942	0.835699	1678.16
14	5000	100	0.984944	0.921579	0.966963	0.833413	1678.16
15	1000	100	0.987018	0.908197	0.947883	0.646069	1678.16
16	1000	200	0.989929	0.928192	0.959082	0.717311	1678.16
17	2000	200	0.993989	0.949445	0.984135	0.869552	1678.16
18	5000	200	0.984925	0.940317	0.966723	0.871213	1678.16
19	10000	200	0.997443	0.941439	0.990374	0.785694	1678.16
20	10000	300	0.995602	0.923871	0.988088	0.801229	1678.16
21	10000	400	0.986459	0.933809	0.970206	0.858222	1678.16
22	10000	1000	0.953615	0.884144	0.831262	0.593529	1678.16
23	10000	1500	0.988821	0.921476	0.970204	0.797798	1678.16
24	10000	2000	0.998812	0.941151	0.997369	0.873415	1678.16
25	10000	5500	0.871685	0.812581	0.509662	0.306422	1678.16
26	10000	6000	0.913386	0.807328	0.775507	0.528297	1678.16

Figure 1 Metrics results for different data set sizes.

Figure 1, we present the results we obtain in terms of R^2 with and without DP. The epsilon column represents the privacy budget we reached. It's important to keep in mind that it's not because the latter is very important that we are not ensuring privacy.

Highlighted in green we have the rows with the highest non-DP R^2 (above .98). We can see that a model infused with differential privacy reacts very well and preserve the level of accuracy the non-DP model has reached. For example, line 17, there's a R^2 of .99 for a DP R^2 of .94; the latter model is still acceptable.
We can conclude that the level of accuracy is retained between a non-DP accurate model and a DP model.

In addition, it's interesting to point out that there isn't any outlier when one compares the non-DP metrics and the DP ones, regardless the level of accuracy reached. Line 1 to 12, we can see that a non-accurate model without DP we'll still behave the same way with DP, (e.g. line 10).
Thus, from this experiment, we can conclude that empirically, behaviors of non-DP and DP models remain the same.