数据隐私实验报告

PB17111614
王嵘晟

DP_SGD

实验结果

当 DP 和 LR 的参数按照代码初始值不改变时:

```
eps, delta = 1.25, 10**(-3)                    # parameters for DP
N, T = 100, 100
w = np.array([[0.5],[0.5]])                    # parameters for LR
```

运行结果:

DP	FM
[[809.79268967]]	[[17.24925647]]
[[24.31224611]]	[[125.44718072]]
[[44.38310162]]	[[64.0052515]]

当 DP 和 LR 选取以下参数时:

```
eps, delta = 1.5, 10**(-3)                    # parameters for DP
N, T = 100, 100
w = np.array([[0.75],[0.25]])                    # parameters for LR
```

运行结果:

DP	FM
[[276.23739514]]	[[16.92727102]]
[[29.63125879]]	[[44.05166569]]
[[1310.39157147]]	[[8.70873894]]

当 DP 和 LR 选取以下参数时:

```
eps, delta = 1.75, 10**(-4)                    # parameters for DP
N, T = 100, 100
w = np.array([[0.25],[0.75]])                    # parameters for LR
```

运行结果:

DP	FM
[[1027.30043606]]	[[4.68036908]]
[[575.65504838]]	[[25.4152409]]
[[146.96517483]]	[[2363.86545063]]

实验结论

通过选取不同参数比较，可以发现整体来看， DP 的 loss 变化范围要大于 FM，显然 FM 更加稳定。但通过选取合适的参数，可以使得 DP 的 loss 比 FM 小。

paillier

实验结果

加法:

位数	m_1	m_2	结果	运行时间
10	640	938	0:00:00.021939	
20	1048575	998453	2047028	0:00:00.019945
50	1125899906842624	990054512268453	2115954419111077	0:00:00.032912
100	1267650600228229401496703205375	111111990054512200000068456453	137876259028274160149677166182 8	0:00:00.012003
200	1.606×10^{60}	9.8×10^{50}	$1.60600000098\times10^{60}$	0:00:00.033898
500	2.561×10^{120}	9.8×10^{100}	$2.561000000000000000098\times10^{122}$	0:00:00.021975
1000	6×10^{240}	720000000000004616846265956230000000	6×10^{241}(约等于)	0:00:00:020979

与明文常数加法:

位数	m_1	m_2	结果	运行时间
10	640	938	1578	0:00:01.697907
20	1048575	998453	2047028	0:00:08.242209

更高位数的内存不足，因此不再往下测试明文常数加法
与明文常数乘法:

位数	m_1	m_2	结果	运行时间
10	640	938	600320	0:00:00.022937
20	1048575	998435	1046933980125	0:00:18.455462

更高位数的内存不足，因此不再往下测试明文常数加法

实验结论

经过验证发现使用 paillier 加密时，满足加法同态的性质，即两个明文加密后的密文相加再解密得到的明文值与两个明文直接相加后与公钥中的n取模的结果相同。密文与明文常数的加法/乘法结果有相似的性质。同时这种加密方法并没有因为明文数据长度的增长导致加密消耗时间增长过多，是一种具有高效性的加密方法。