

多方场景下纵向逻辑回归模型

这里假设是两方来联合训练一个模型。在联邦学习中，有以下几种角色：

guest:数据应用方，在纵向FL中指的是含有数据标签的一方。而且建模流程一般由guest方发起

host: 数据的提供方

arbiter:主要来辅助多方联合建模的。纵向FL中用来分发公私钥、加解密服务等等。它本身既不提供数据，也不使用数据

多方场景一：纵向逻辑回归（使用同态加密来保护数据隐私）

首先要明确最终的目标：最终我要得到两个子模型，对于guest方(为了公式描述方便，以下也叫做A方)而言，我要得到子模型 w_A ；对于host方而言，我要得到子模型(为了公式描述方便，以下也叫做B方) w_B

这也意味着当我使用模型进行预测的时候，我需要联合两个子模型 w_A 和 w_B 才可以进行预测

损失函数

$$F(\omega) = \frac{1}{n} \sum_{n=1}^n \log(1 + e^{-y_i \omega^T x_i})$$

这里损失函数形式和单方场景下略有区别，多方交互的联邦学习中逻辑回归函数都是这个形式的损失函数。我也不太清楚这样的好处

上述损失函数的梯度公式是

$$\nabla F(\omega) = \frac{1}{n} \sum_{n=1}^n \left(\frac{1}{1 + e^{-y_i \omega^T x_i}} - 1 \right) y_i x_i$$

由于同态加密不支持指数运算，所以需要将上述的指数形式近似转化为**多项式**才可以使用同态加密技术

所以损失函数的梯度公式可以**近似的**写成

$$\nabla F(\omega) \approx \frac{1}{n} \sum_{n=1}^n (0.25 \omega^T x_i - 0.5 y_i) x_i = \frac{1}{n} \sum_{n=1}^n d_i x_i$$

在纵向联邦学习（使用同态加密技术）的场景下，guest方拥有数据 X_A ，以及数据标签 Y 。host方(为了公式描述方便，以下也叫做B方)拥有数据 X_B

所以完整的 $\omega x = \omega_A X_A + \omega_B X_B$

这里有一个符号表示： $\omega_A X_A$ 当使用同态加密形式，则表示为：
[[$\omega_A X_A$]]

算法执行流程

1. arbiter方分发公钥给A方和B方
2. B方计算 $\omega_B X_B$ ，然后加密为 [[$\omega_B X_B$]]。并且发给A方
3. A方收到 [[$\omega_B X_B$]] 后，同时自身也计算 [[$\omega_A X_A$]]。然后获得完整的：[[ωx]] = [[$\omega_A X_A$]] + [[$\omega_B X_B$]]
4. 这样，A方便获得了完整的关于每条数据的 d_i 。然后，将 d_i 发送给B方

经过上述的4个步骤，双方便分别确定了下式（密文状态下）：

对于A方而言：

$$\nabla F(\omega)_A = \frac{1}{n} \sum_{n=1}^n (0.25 \omega^T x - 0.5 y) x_A = \frac{1}{n} \sum_{n=1}^n d_i x_A$$

对于B方而言：

$$\nabla F(\omega)_B = \frac{1}{n} \sum_{n=1}^n (0.25\omega^T x - 0.5y)x_B = \frac{1}{n} \sum_{n=1}^n d_i x_B$$

5. A、B双方求出梯度（密文）后，开始发送给arbiter方。然后它来解密梯度，再发送给A方和B方
6. A方、B方收到解密的梯度后，开始更新模型参数 ω_A 和 ω_B
7. 重复上述2~6过程，直到收敛或者达到指定的迭代次数

隐私泄露风险

当使用同态加密的技术的时候，A方在当前批出会获得关于B的系列 $w_B^T x_B$ 。B方会获得关于A的 $h(w_i, x_i, y_i)$ 系列信息。加入参与方存在攻击者的话，容易从中窃取用户隐私。已有论文已经证明这一点了

论文参考：

《exploiting unintended feature leakage in collaborative learning》

《model inversion attacks that exploit confidence information and basic countermeasures》

多方场景二：纵向逻辑回归（使用差分隐私技术去保护隐私）

这篇文章的参考链接如下：

Hybrid Differentially Private Federated Learning on Vertically Partitioned Data

上述这篇文章有待于优化的就是全局敏感度容易太大，导致噪声太大，致使模型训练效果不太理想。这篇文章的主要思想如下：

预处理：将数据归一化到-1~1之间。然后开始如下迭代过程：

1. B方选择小批量数据集 X_t (t 代表当前的迭代次数)，然后计算

$$IR_t^B = X_t^B \omega_t^B$$

添加噪声，公式为： $Sec[IR_t^B] = IR_t^B + Z^B$ 。这样， $Sec[IR_t^B]$ 就是添加噪声后的 IR_t^B 。而后将此结果发送给A方

疑问点：这里的 IR_t^B 是否是取均值后的结果，或者它就是当前批次的若干条数据。

我这里是当作**当前批次的若干条数据**来处理，给每条数据添加噪音。

2. 这样，A方就获得了完整的 $\omega x = \omega_A X_A + \omega_B X_B$ ，只不过这里的 $\omega_B X_B$ 添加了噪音。根据逻辑回归的整体损失函数，可以表示成(b 代表的是当前的小批量)：

$$\nabla F(\omega) = \sum_{n=1}^b \left(\frac{y_i}{1 + e^{-y_i \omega^T x_i}} - y_i \right) x_i = \sum_{n=1}^b d_i x_i$$

在这里将 d_i 表示成 IR_t^A 。然后 IR_t^A 添加噪声， $Sec[IR_t^A] = IR_t^A + Z_A$ 。将这个结果发送给B方

3. 经过一轮交互之后，A方、B方各自获得：

$$\nabla F(\omega)_A = \sum_{n=1}^b Sec[IR_t^A] x_A$$

$$\nabla F(\omega)_B = \sum_{n=1}^b Sec[IR_t^B] x_B$$

4. 下面的就和常规的逻辑回归一样了，A、B双方得到各自的梯度，于是更新子模型： ω_A 和 ω_B

5. 重复1-4步骤便可。直到收敛或者达到指定的迭代次数

基于HE和DP技术的纵向联邦学习模型

本项目的完整GitHub地址如下 (<https://github.com/XtangEver/HDP-VFL>) :

[项目github地址](#)

B方添加的噪音

当A方计算出密文梯度:

$$\nabla F(\omega)_A = \frac{1}{n} \sum_{n=1}^n (0.25\omega^T x - 0.5y)x_A = \frac{1}{n} \sum_{n=1}^n d_i x_A$$

需要将上述结果发送给B方，B方采用DP技术添加噪音（高斯噪声），其中全局敏感度的公式如下：

$$\Delta f = \sqrt{\frac{4L^2 e^2 T \eta^2}{b} + \frac{8kLe^2 \eta}{b} + 4k^2 e}$$

在上述结果的基础上，我们假设所有的数据都已正则化到-1~1之间。那么任意改变一条记录，所会改变的数据的L2敏感度是： $\sqrt{(n)}$ 。其中n代表B方数据的维度信息，所以B所添加的噪音对应的全敏感度公式是：

$$\Delta f = \sqrt{\frac{4L^2 e^2 T \eta^2}{b} + \frac{8kLe^2 \eta}{b} + 4k^2 e} * \sqrt{n}/b$$

其中参数所对应的含义如下：

```
def
gaussian(self,delta,epsilon,L,e,T,eta,b,k,length):
    """
    生成高斯噪声所需要的loc、sigma_a
    parameters
```

DP机制

delta:一定程度的允许错误的值，因为高斯机制非严格满足

epsilon:隐私保护预算

L:lipschitz 常数，默认值为1

e:epochs

T: $e * r$,表示总的迭代次数

eta:learning rate

b:mini-batch size

k:梯度剪切参数

length:表述数据维度的长度

loc, sigma都是高斯分布的俩参数

"""

loc = 0

partial_1 = $\text{np.sqrt}(2 * \text{np.log}(1.25 / \text{delta}))$

partial_2_1 = $(4 * \text{np.square}(L) * \text{np.square}(e) * T * \text{np.square}(\text{eta})) / b$

partial_2_2 = $(8 * k * L * \text{np.square}(e) * \text{eta}) / b$

partial_2_3 = $4 * \text{np.square}(k) * e$

partial_2 = $\text{np.sqrt}(\text{partial_2_1} + \text{partial_2_2} + \text{partial_2_3}) * \text{np.sqrt}(\text{length}) / b$ #最大敏感度

partial_3 = epsilon

sigma_a = $\text{partial_1} * \text{partial_2} / \text{partial_3}$

LOGGER.info("站在**host**方，全局敏感度值：
{ }".format(**partial_2**))

LOGGER.info("站在**host**方，标准差是：
{ }".format(**sigma_a**))

return loc, sigma_a

A方添加的噪音

当B计算出密文梯度后:

$$\nabla F(\omega)_B = \frac{1}{n} \sum_{n=1}^n (0.25\omega^T x - 0.5y)x_B = \frac{1}{n} \sum_{n=1}^n d_i x_B$$

需要将上述结果发送给A方，A方采用DP技术添加噪音（高斯噪声），其中全局敏感度的公式如下

$$\Delta f = \sqrt{4\beta_\theta^2 L^2 \frac{e^2 T \eta^2}{b} + 8(\beta_\theta k + \beta_y k_y) \beta_\theta L \frac{e^2 \eta}{b} + 4(\beta_\theta k + \beta_y k_y)^2 e}$$

当任意改变A中数据的一条记录，所会改变的数据的L2敏感度是 $\sqrt{(n)}$ 。其中n代表A方数据的维度信息，所以B所添加的噪音对应的全敏感度公式是：

$$\Delta f = \sqrt{4\beta_\theta^2 L^2 \frac{e^2 T \eta^2}{b} + 8(\beta_\theta k + \beta_y k_y) \beta_\theta L \frac{e^2 \eta}{b} + 4(\beta_\theta k + \beta_y k_y)^2 e} * \sqrt{n}/b$$

其中若干参数所对应的含义是：

```
def
gaussian(self,delta,epsilon,beta_theta,L,e,T,eta,b,k
,beta_y,k_y,length):
    """
    生成高斯噪声所需要的loc、sigma_b
    parameters
    -----
    delta:非严格的DP损失
    epsilon:隐私保护预算
    beta_theta:smooth parameters,未知
    L:lipschitz 常数，默认值为1
    e:epochs
    T:e * r,表示总的迭代次数
    eta:learning rate
    b:mini-batch size
```

```

k:梯度剪切参数
beta_y:smooth parameters,未知
k_y:target bound,未知
length:host方数据的维度，作为全局敏感度的一个参数
"""

loc = 0
partial_1 = np.sqrt(2 * np.log(1.25 /
delta))

partial_2_1 = 4 * np.square(beta_theta) *
np.square(L) * np.square(e) * T * np.square(eta) / b
partial_2_2 = 8 * (beta_theta * k + beta_y *
k_y) * beta_theta * L * np.square(e) * eta / b
partial_2_3 = 4 * np.square(beta_theta * k +
beta_y * k_y) * e
partial_2 = np.sqrt(partial_2_1 +
partial_2_2 + partial_2_3) * np.sqrt(length) / b #最大敏感度

partial_3 = epsilon

sigma_b = partial_1 * partial_2 / partial_3
LOGGER.info("在guest方，全局敏感度值：
{}".format(partial_2))
LOGGER.info("在guest方，标准差sigma_b:
{}".format(sigma_b))
return loc,sigma_b

```

实验结果如下

当不添加噪音的时候，只使用HE技术：

Evaluation Scores

Quantile: - 0.5 + ?

	dataset	auc
hetero_lr_0	train	0.995588
hetero_lr_1	predict	0.982903

Confusion Matrix

Classification Threshold: - 0.5 + ?

	dataset	F1-score
hetero_lr_1	predict	0.974093
hetero_lr_0	train	0.972171

当添加高斯噪声后

Evaluation scores

Quantile: - 0.5 + ?

	dataset	auc
hdp_vfl_0	train	0.967029
hdp_vfl_1	predict	0.964733

Confusion Matrix

Classification Threshold: - 0.5 + ?

	dataset	F1-score
hdp_vfl_1	predict	0.921053
hdp_vfl_0	train	0.933786

其中A方以及B方的全局敏感度以及噪音如下：

```
3130 [Info] [2021-10-14 22:00:22,892] [33785:140030594897728] - linear_model_weight.py[line:361]: 站在host方，标准差是:1.0361585612531785
3131 [Info] [2021-10-14 22:00:22,892] [33785:140030594897728] - hdp_vfl_host.py[line:118]: 开始对guest侧数据添加噪声
3132 [Info] [2021-10-14 22:00:22,892] [33785:140030594897728] - linear_model_weight.py[line:289]: 添加的噪音为: [-0.0441479825737605, 0.5684539353446176, -0.05581664095311943, -1.951100840491126, -0.4394004061250489,
0.005123589838118548, 0.9868841563257778, -0.3532288485569002, 0.4023344871103769, 0.550460551638315, -0.9844514820901599]

3733 [Info] [2021-10-14 22:00:24,099] [33783:140655634892608] - linear_model_weight.py[line:146]: 在guest方，标准差sigma_b:1.6883934808210093
3734 [Info] [2021-10-14 22:00:24,100] [33783:140655634892608] - hdp_vfl_guest.py[line:143]: 开始对host侧数据添加噪声
3735 [Info] [2021-10-14 22:00:24,100] [33783:140655634892608] - linear_model_weight.py[line:164]: 添加的噪音为: [-0.5920334060221425, -0.5167877381086938, -1.4025250136614813, -0.9650085592889986, 1.9369610011773724,
-2.462077192149632, -2.672945993593817, 1.2390816657597727, 0.14935812851018096, 1.1992074732313225, 1.2897844164403378, -0.802334870308512, -2.230099079345016, -2.703305073943483, 1.7430286452162675, -0.47738
047405502154, 2.0813319087443456, 1.7425863442117997, 1.6526795732183834, 0.6917848833231898]

...
Process: 33783, 33784, 33785, 33786, 33787, 33788, 33789, 33790, 33791, 33792, 33793, 33794, 33795, 33796, 33797, 33798, 33799, 33800, 33801, 33802, 33803, 33804, 33805, 33806, 33807, 33808, 33809, 33810, 33811, 33812, 33813, 33814, 33815, 33816, 33817, 33818, 33819, 33820, 33821, 33822, 33823, 33824, 33825, 33826, 33827, 33828, 33829, 33830, 33831, 33832, 33833, 33834, 33835, 33836, 33837, 33838, 33839, 33840, 33841, 33842, 33843, 33844, 33845, 33846, 33847, 33848, 33849, 33850, 33851, 33852, 33853, 33854, 33855, 33856, 33857, 33858, 33859, 33860, 33861, 33862, 33863, 33864, 33865, 33866, 33867, 33868, 33869, 33870, 33871, 33872, 33873, 33874, 33875, 33876, 33877, 33878, 33879, 33880, 33881, 33882, 33883, 33884, 33885, 33886, 33887, 33888, 33889, 33890, 33891, 33892, 33893, 33894, 33895, 33896, 33897, 33898, 33899, 33900, 33901, 33902, 33903, 33904, 33905, 33906, 33907, 33908, 33909, 33910, 33911, 33912, 33913, 33914, 33915, 33916, 33917, 33918, 33919, 33920, 33921, 33922, 33923, 33924, 33925, 33926, 33927, 33928, 33929, 33930, 33931, 33932, 33933, 33934, 33935, 33936, 33937, 33938, 33939, 33940, 33941, 33942, 33943, 33944, 33945, 33946, 33947, 33948, 33949, 33950, 33951, 33952, 33953, 33954, 33955, 33956, 33957, 33958, 33959, 33960, 33961, 33962, 33963, 33964, 33965, 33966, 33967, 33968, 33969, 33970, 33971, 33972, 33973, 33974, 33975, 33976, 33977, 33978, 33979, 33980, 33981, 33982, 33983, 33984, 33985, 33986, 33987, 33988, 33989, 33990, 33991, 33992, 33993, 33994, 33995, 33996, 33997, 33998, 33999, 34000, 34001, 34002, 34003, 34004, 34005, 34006, 34007, 34008, 34009, 34010, 34011, 34012, 34013, 34014, 34015, 34016, 34017, 34018, 34019, 34020, 34021, 34022, 34023, 34024, 34025, 34026, 34027, 34028, 34029, 34030, 34031, 34032, 34033, 34034, 34035, 34036, 34037, 34038, 34039, 34040, 34041, 34042, 34043, 34044, 34045, 34046, 34047, 34048, 34049, 34050, 34051, 34052, 34053, 34054, 34055, 34056, 34057, 34058, 34059, 34060, 34061, 34062, 34063, 34064, 34065, 34066, 34067, 34068, 34069, 34070, 34071, 34072, 34073, 34074, 34075, 34076, 34077, 34078, 34079, 34080, 34081, 34082, 34083, 34084, 34085, 34086, 34087, 34088, 34089, 34090, 34091, 34092, 34093, 34094, 34095, 34096, 34097, 34098, 34099, 34100, 34101, 34102, 34103, 34104, 34105, 34106, 34107, 34108, 34109, 34110, 34111, 34112, 34113, 34114, 34115, 34116, 34117, 34118, 34119, 34120, 34121, 34122, 34123, 34124, 34125, 34126, 34127, 34128, 34129, 34130, 34131, 34132, 34133, 34134, 34135, 34136, 34137, 34138, 34139, 34140, 34141, 34142, 34143, 34144, 34145, 34146, 34147, 34148, 34149, 34150, 34151, 34152, 34153, 34154, 34155, 34156, 34157, 34158, 34159, 34160, 34161, 34162, 34163, 34164, 34165, 34166, 34167, 34168, 34169, 34170, 34171, 34172, 34173, 34174, 34175, 34176, 34177, 34178, 34179, 34180, 34181, 34182, 34183, 34184, 34185, 34186, 34187, 34188, 34189, 34190, 34191, 34192, 34193, 34194, 34195, 34196, 34197, 34198, 34199, 34200, 34201, 34202, 34203, 34204, 34205, 34206, 34207, 34208, 34209, 34210, 34211, 34212, 34213, 34214, 34215, 34216, 34217, 34218, 34219, 34220, 34221, 34222, 34223, 34224, 34225, 34226, 34227, 34228, 34229, 34230, 34231, 34232, 34233, 34234, 34235, 34236, 34237, 34238, 34239, 34240, 34241, 34242, 34243, 34244, 34245, 34246, 34247, 34248, 34249, 34250, 34251, 34252, 34253, 34254, 34255, 34256, 34257, 34258, 34259, 34260, 34261, 34262, 34263, 34264, 34265, 34266, 34267, 34268, 34269, 34270, 34271, 34272, 34273, 34274, 34275, 34276, 34277, 34278, 34279, 34280, 34281, 34282, 34283, 34284, 34285, 34286, 34287, 34288, 34289, 34290, 34291, 34292, 34293, 34294, 34295, 34296, 34297, 34298, 34299, 34300, 34301, 34302, 34303, 34304, 34305, 34306, 34307, 34308, 34309, 34310, 34311, 34312, 34313, 34314, 34315, 34316, 34317, 34318, 34319, 34320, 34321, 34322, 34323, 34324, 34325, 34326, 34327, 34328, 34329, 34330, 34331, 34332, 34333, 34334, 34335, 34336, 34337, 34338, 34339, 34340, 34341, 34342, 34343, 34344, 34345, 34346, 34347, 34348, 34349, 34350, 34351, 34352, 34353, 34354, 34355, 34356, 34357, 34358, 34359, 34360, 34361, 34362, 34363, 34364, 34365, 34366, 34367, 34368, 34369, 34370, 34371, 34372, 34373, 34374, 34375, 34376, 34377, 34378, 34379, 34380, 34381, 34382, 34383, 34384, 34385, 34386, 34387, 34388, 34389, 34390, 34391, 34392, 34393, 34394, 34395, 34396, 34397, 34398, 34399, 34400, 34401, 34402, 34403, 34404, 34405, 34406, 34407, 34408, 34409, 34410, 34411, 34412, 34413, 34414, 34415, 34416, 34417, 34418, 34419, 34420, 34421, 34422, 34423, 34424, 34425, 34426, 34427, 34428, 34429, 34430, 34431, 34432, 34433, 34434, 34435, 34436, 34437, 34438, 34439, 34440, 34441, 34442, 34443, 34444, 34445, 34446, 34447, 34448, 34449, 34450, 34451, 34452, 34453, 34454, 34455, 34456, 34457, 34458, 34459, 34460, 34461, 34462, 34463, 34464, 34465, 34466, 34467, 34468, 34469, 34470, 34471, 34472, 34473, 34474, 34475, 34476, 34477, 34478, 34479, 34480, 34481, 34482, 34483, 34484, 34485, 34486, 34487, 34488, 34489, 34490, 34491, 34492, 34493, 34494, 34495, 34496, 34497, 34498, 34499, 34500, 34501, 34502, 34503, 34504, 34505, 34506, 34507, 34508, 34509, 34510, 34511, 34512, 34513, 34514, 34515, 34516, 34517, 34518, 34519, 34520, 34521, 34522, 34523, 34524, 34525, 34526, 34527, 34528, 34529, 34530, 34531, 34532, 34533, 34534, 34535, 34536, 34537, 34538, 34539, 34540, 34541, 34542, 34543, 34544, 34545, 34546, 34547, 34548, 34549, 34550, 34551, 34552, 34553, 34554, 34555, 34556, 34557, 34558, 34559, 34560, 34561, 34562, 34563, 34564, 34565, 34566, 34567, 34568, 34569, 34570, 34571, 34572, 34573, 34574, 34575, 34576, 34577, 34578, 34579, 34580, 34581, 34582, 34583, 34584, 34585, 34586, 34587, 34588, 34589, 34590, 34591, 34592, 34593, 34594, 34595, 34596, 34597, 34598, 34599, 34600, 34601, 34602, 34603, 34604, 34605, 34606, 34607, 34608, 34609, 34610, 34611, 34612, 34613, 34614, 34615, 34616, 34617, 34618, 34619, 34620, 34621, 34622, 34623, 34624, 34625, 34626, 34627, 34628, 34629, 34630, 34631, 34632, 34633, 34634, 34635, 34636, 34637, 34638, 34639, 34640, 34641, 34642, 34643, 34644, 34645, 34646, 34647, 34648, 34649, 34650, 34651, 34652, 34653, 34654, 34655, 34656, 34657, 34658, 34659, 34660, 34661, 34662, 34663, 34664, 34665, 34666, 34667, 34668, 34669, 34670, 34671, 34672, 34673, 34674, 34675, 34676, 34677, 34678, 34679, 34680, 34681, 34682, 34683, 34684, 34685, 34686, 34687, 34688, 34689, 34690, 34691, 34692, 34693, 34694, 34695, 34696, 34697, 34698, 34699, 34700, 34701, 34702, 34703, 34704, 34705, 34706, 34707, 34708, 34709, 34710, 34711, 34712, 34713, 34714, 34715, 34716, 34717, 34718, 34719, 34720, 34721, 34722, 34723, 34724, 34725, 34726, 34727, 34728, 34729, 34730, 34731, 34732, 34733, 34734, 34735, 34736, 34737, 34738, 34739, 34740, 34741, 34742, 34743, 34744, 34745, 34746, 34747, 34748, 34749, 34750, 34751, 34752, 34753, 34754, 34755, 34756, 34757, 34758, 34759, 34760, 34761, 34762, 34763, 34764, 34765, 34766, 34767, 34768, 34769, 34770, 34771, 34772, 34773, 34774, 34775, 34776, 34777, 34778, 34779, 34780, 34781, 34782, 34783, 34784, 34785, 34786, 34787, 34788, 34789, 34790, 34791, 34792, 34793, 34794, 34795, 34796, 34797, 34798, 34799, 34800, 34801, 34802, 34803, 34804, 34805, 34806, 34807, 34808, 34809, 34810, 34811, 34812, 34813, 34814, 34815, 34816, 34817, 34818, 34819, 34820, 34821, 34822, 34823, 34824, 34825, 34826, 34827, 34828, 34829, 34830, 34831, 34832, 34833, 34834, 34835, 34836, 34837, 34838, 34839, 34840, 34841, 34842, 34843, 34844, 34845, 34846, 34847, 34848, 34849, 34850, 34851, 34852, 34853, 34854, 34855, 34856, 34857, 34858, 34859, 34860, 34861, 34862, 34863, 34864, 34865, 34866, 34867, 34868, 34869, 34870, 34871, 34872, 34873, 34874, 34875, 34876, 34877, 34878, 34879, 34880, 34881, 34882, 34883, 34884, 34885, 34886, 34887, 34888, 34889, 34890, 34891, 34892, 34893, 34894, 34895, 34896, 34897, 34898, 34899, 34900, 34901, 34902, 34903, 34904, 34905, 34906, 34907, 34908, 34909, 34910, 34911, 34912, 34913, 34914, 34915, 34916, 34917, 34918, 34919, 34920, 34921, 34922, 34923, 34924, 34925, 34926, 34927, 34928, 34929, 34930, 34931, 34932, 34933, 34934, 34935, 34936, 34937, 34938, 34939, 34940, 34941, 34942, 34943, 34944, 34945, 34946, 34947, 34948, 34949, 34950, 34951, 34952, 34953, 34954, 34955, 34956, 34957, 34958, 34959, 34960, 34961, 34962, 34963, 34964, 34965, 34966, 34967, 34968, 34969, 34970, 34971, 34972, 34973, 34974, 34975, 34976, 34977, 34978, 34979, 34980, 34981, 34982, 34983, 34984, 34985, 34986, 34987, 34988, 34989, 34990, 34991, 34992, 34993, 34994, 34995, 34996, 34997, 34998, 34999, 35000, 35001, 35002, 35003, 35004, 35005, 35006, 35007, 35008, 35009, 35010, 35011, 35012, 35013, 35014, 35015, 35016, 35017, 35018, 35019, 35020, 35021, 35022, 35023, 35024, 35025, 35026, 35027, 35028, 35029, 35030, 35031, 35032, 35033, 35034, 35035, 35036, 35037, 35038, 35039, 35040, 35041, 35042, 35043, 35044, 35045, 35046, 35047, 35048, 35049, 35050, 35051, 35052, 35053, 35054, 35055, 35056, 35057, 35058, 35059, 35060, 35061, 35062, 35063, 35064, 35065, 35066, 35067, 35068, 35069, 35070, 35071, 35072, 35073, 35074, 35075, 35076, 35077, 35078, 35079, 35080, 35081, 35082, 35083, 35084, 35085, 35086, 35087, 35088, 35089, 35090, 35091, 35092, 35093, 35094, 35095, 35096, 35097, 35098, 35099, 35100, 35101, 35102, 35103, 35104, 35105, 35106, 35107, 35108, 35109, 35110, 35111, 35112, 35113, 35114, 35115, 35116, 35117, 35118, 35119, 35120, 35121, 35122, 35123, 35124, 35125, 35126, 35127, 35128, 35129, 35130, 35131, 35132, 35133, 35134, 35135, 35136, 35137, 35138, 35139, 35140, 35141, 35142, 35143, 35144, 35145, 35146, 35147, 35148, 35149, 35150, 35151, 35152, 35153, 35154, 35155, 35156, 35157, 35158, 35159, 35160, 35161, 35162, 35163, 35164, 35165, 35166, 35167, 35168, 35169, 35170, 35171, 35172, 35173, 35174, 35175, 35176, 35177, 35178, 35179, 35180, 35181, 35182, 35183, 35184, 35185, 35186, 35187, 35188, 35189, 35190, 35191, 35192, 35193, 35194, 35195, 35196, 35197, 35198, 35199, 35200, 35201, 35202, 35203, 35204, 35205, 35206, 35207, 35208, 35209, 35210, 35211, 35212, 35213, 35214, 35215, 35216, 35217, 35218, 35219, 35220, 35221, 35222, 35223, 35224, 35225, 35226, 35227, 35228, 35229, 35230, 35231, 35232, 35233, 35234, 35235, 35236, 35237, 35238, 35239, 35240, 35241, 35242, 35243, 35244, 35245, 35246, 35247, 35248, 35249, 35250, 35251, 35252, 35253, 35254, 35255,
```

总结

我们牺牲了非常小的精度，却换取了用户的隐私的保护强度提升。
是很值得的！