基于生成对抗网络的差分隐私数据发布方法

方 晨[1],郭渊博[1],王 娜[1],甄帅辉[1,2],唐国栋[3]
(1. 信息工程大学,河南郑州 450001; 2. 中国人民解放军 93808 部队,甘肃兰州 730000;
3. 中国人民解放军 75775 部队,广东广州 510000)

摘 要: 机器学习的飞速发展使其成为数据挖掘领域最有效的工具之一,但算法的训练过程往往需要大量的用户数据,给用户带来了极大的隐私泄漏风险. 由于数据统计特征的复杂性及语义丰富性,传统隐私数据发布方法往往需要对原始数据进行过度清洗,导致数据可用性低而难以再适用于数据挖掘任务. 为此,提出了一种基于生成对抗网络(Generative Adversarial Network,GAN)的差分隐私数据发布方法,通过在 GAN 模型训练的梯度上添加精心设计的噪声来实现差分隐私,确保 GAN 可无限量生成符合源数据统计特性且不泄露隐私的合成数据. 针对现有同类方法合成数据质量低、模型收敛缓慢等问题,设计多种优化策略来灵活调整隐私预算分配并减小总体噪声规模,同时从理论上证明了合成数据严格满足差分隐私特性. 在公开数据集上与现有方法进行实验对比,结果表明本方法能够更高效地生成质量更高的隐私保护数据,适用于多种数据分析任务.

关键词: 差分隐私; 生成对抗网络; 隐私数据发布; 合成数据; 数据挖掘
中图分类号: TP301 **文献标识码:** A **文章编号:** 0372-2112 (2020)10-1983-10
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2020.10.016

Differential Private Data Publishing Method Based on Generative Adversarial Network

FANG Chen[1],GUO Yuan-bo[1],WANG Na[1],ZHEN Shuai-hui[1,2],TANG Guo-dong[3]
(1. *Information Engineering University*,*Zhengzhou*,*Henan* 450001,*China*;
2. *Unit* 93808,*Lanzhou*,*Gansu* 730000,*China*;3. *Unit* 75775,*Guangzhou*,*Guangdong* 510000,*China*)

Abstract: The rapid development of machine learning makes itself one of the most effective tools in the data mining research community. However,the training of algorithm often needs a large amount of user data,which brings a great risk of privacy leakage to users. Due to the complex statistical characteristics and semantic richness of the data,traditional private data publishing methods usually sanitize original data too excessively to lead to low data availability and uselessness in data mining tasks. In this paper,a differential private data publishing method based on generative adversarial network (GAN) is proposed. The differential privacy of the GAN model is realized by adding carefully designed noise to the gradients during the training procedure,so that the GAN can generate unlimited synthetic data conforming to the original statistical characteristics without disclosing any privacy. Aiming at the problems of low quality synthetic data and slow convergence in the existing similar methods,several optimization strategies are designed to adjust the privacy budget allocation and reduce the overall noise scale. Moreover,we provide rigorous proof that the synthetic data satisfies the differential privacy. Comparisons with existing methods on public datasets show that the method proposed can generate private data with higher quality more efficiently,which is suitable for various data analysis tasks.

Key words: differential privacy;generative adversarial network;private data publishing;synthetic data;data mining

1 引言

随着移动计算的发展和多媒体网络的普及,各类信息系统收集了大量语义丰富的用户数据. 人们研究出多种机器学习算法来挖掘数据中有价值的信息,以此提供各式各样的服务,如个性化商品推荐、广告投放等. 但与此同时也出现了严重的隐私泄露问题. 已有研究[1]表明攻击者能够利用训练好的神经网络模型推断

收稿日期:2019-12-10;修回日期:2020-04-16;责任编辑:孙瑶
基金项目: 国家自然科学基金(No. 61501515,No. 61601515);信息保障技术重点实验室开放基金(No. KJ-15-108)

出用户的隐私信息,如就诊记录、银行资金流水等. 因此数据挖掘应在隐私保护的前提下进行. 现有的隐私保护方法分为交互式和非交互式两种[2]. 交互式方法是指由数据拥有者设计满足隐私保护性质的接口以供分析人员做数据查询,但查询次数过多会导致数据误差较大;非交互式方法是指数据拥有者将原始数据中的敏感信息清洗后再对外发布,能够在保留原始数据效用的同时保护用户的隐私,是当前隐私保护领域的研究热点.

k-匿名、l-多样化、t-closeness 等是最早提出的隐私保护方法,这类方法能够较好地保护数据细节,但它们依赖于对手的背景知识假设,且难以抵抗新出现的组合攻击、前景知识攻击等手段[3]. 差分隐私通过在统计结果上添加适量的噪声来确保任何一条记录的修改都不会对整体结果产生显著的影响,具有严格的数学基础和对背景知识的弱依赖[2]. 傅继彬等[4]提出基于差分隐私的决策树数据发布方法,通过最大值属性选择算法进行层次细化,并利用类几何策略更合理地分配隐私预算. 但是该方法的计算开销和添加的噪声会随着数据维度的增加而变大,导致合成数据集可用性不高. 为此,Zhang 等[5]提出了 PrivBayes 方法,基于贝叶斯网络将高维数据集转化为低维数据集,并利用差分隐私技术实现了高维数据集的安全发布. 为了提高数据发布的效率,Asghar 等[6]使用高斯 copula 函数定义输入数据集中属性的依赖关系,将这些属性的行建模为多元分布的样本,然后通过 copula 对目标分布进行采样来合成数据集. 但是,上述方法均是针对于结构化数据集. 近几年随着生成对抗网络(Generative Adversarial Network,GAN)[7]在计算机视觉领域的兴起,有学者开始考虑将其与差分隐私相结合. 基于 GAN 对于原始数据复杂特征的学习能力,在差分隐私的条件下训练 GAN,可以无限量生成高效用且满足差分隐私特性的合成数据,同时能够缓解部分场景下高质量数据匮乏的问题. Xie 等[8]提出差分隐私 GAN 的通用方法,通过在判别器的梯度中添加噪声来保证生成数据满足差分隐私,但该方法并未提出任何优化策略来改进训练过程的稳定性和模型精度. Acs 等[9]利用差分隐私核 k-means 算法将训练数据划分成 k 个簇,然后在每个簇上单独训练一个生成神经网络,最终将它们混合在一起模拟训练数据的分布. Xu 等[10]提出自适应梯度裁剪策略来减小训练数据的隐私损失,郭鹏等[11]在原始差分隐私 GAN 模型的基础上添加类别标签作为辅助信息,提高了生成数据的质量. 但文献[10]和文献[11]都假设需要有小部分公开数据集,在某些场景下(如医疗数据集)该假设不一定成立.

由此可见,如何在有限的隐私成本下生成高质量的隐私保护数据及确保模型训练过程的稳定性是当前该领域面临的主要挑战. 为此,本文基于一种更稳定的 GAN 变体——WGAN-GP[12],提出一种差分隐私数据发布方法. 通过动态分配隐私预算、自适应选取裁剪阈值和参数分类裁剪等优化策略,提高模型收敛速度和生成隐私保护数据的质量.

本文的贡献有以下几点:

(1)提出适用于差分隐私深度学习算法的动态隐私预算分配策略,在算法训练过程中动态调整噪声规模并利用时刻统计实时追踪隐私损失.

(2)基于自适应裁剪阈值将参数聚类并实施梯度分类裁剪,有效加快了算法的收敛速度.

(3)从理论上证明了合成的数据严格满足差分隐私特性,且隐私损失与合成数据量无关,可应用于大型数据集.

(4)在真实的数据集上对算法进行全面的评估,证明了本方法能够在较小的隐私预算下合成高质量的数据,可用于多种数据分析任务.

2 相关技术

2.1 差分隐私

差分隐私是一种严格可证明的数学框架,其基本思想是对原始数据转换或对输出结果添加噪声来保护数据隐私,确保数据集中任何单个记录的修改都不会对统计结果造成显著的影响. 相关定义如下:

定义 1[2] ((ε,δ)-差分隐私). 令 $A:D\to R$ 为随机算法,D 和 D' 为最多相差一条记录的相邻数据集,若 A 在 D 和 D' 上任意输出结果 $O\in R$ 都满足式(1),则称 A 实现 (ε,δ)-差分隐私.

$$\Pr[A(D)=O]\leqslant e^{\varepsilon}\times\Pr[A(D')=O]+\delta \tag{1}$$

其中参数 ε 为隐私预算,ε 值越小表示隐私保护程度越高;δ 表示违背严格差分隐私的概率.

定义 2[2] (敏感度). 对于任意函数 $f:D\to R^d$,其敏感度为

$$\Delta f=\max_{D,D'}\|f(D)-f(D')\|_p \tag{2}$$

其中 D 和 D' 最多相差一条记录,$\|\bullet\|_p$ 表示 L_p 范数. 敏感度衡量了单条记录对于 f 输出的最大影响,它决定了需要向 f 的输出结果中添加多少噪声来实现差分隐私.

定义 **3**[2] (高斯机制). 当算法 A 的输出结果为数值型时,若用 L_2 范数定义敏感度,可通过向函数 f 的输出中添加高斯噪声来实现差分隐私,如式(3)所示.

$$A(D)=f(D)+N(0,(\Delta f\sigma)^2\boldsymbol{I}) \tag{3}$$

其中 $N(0,(\Delta f\sigma)^2\boldsymbol{I})$ 是均值为 0,方差为 $\Delta f^2\sigma^2$ 的高斯噪声,$\boldsymbol{I}$ 为单位矩阵. 若 $\sigma^2\geqslant 2\log\left(\frac{1.25}{\delta}\right)\Big/\varepsilon^2$ 且 $\varepsilon\in(0,$

1),则算法 A 满足(ε,δ)-差分隐私[2].

定义 4[2]　(指数机制). 当算法 A 的输出结果为非数值型时,给定打分函数 $f:(D\times O)\to R$,若算法 A 满足式(4),则 A 满足 ε-差分隐私.

$$A(D,f)=\left\{r:|\Pr[r\in O]\propto\exp\left(\frac{\varepsilon f(D,r)}{2\Delta f}\right)\right\} \tag{4}$$

2.2　生成对抗网络

受二元零和博弈的启发,GAN 由一对相互对抗的模型组成:生成器 G 和判别器 D. 生成器的输入为噪声,其目标是使噪声分布 p_z 逼近真实数据分布 p_{data};判别器的输入为生成数据与真实数据,其目标是尽可能区分出两种数据. 在博弈过程中两者不断提高各自的生成能力和判别能力,最终达到一个纳什均衡. 其可描述为如下极大极小博弈问题.

$$\min_G\max_D V(D,G)=E_{x\sim p_{\text{data}}(x)}[\log D_w(x)]+E_{z\sim p_z(z)}[\log(1-D_w(G_\theta(z)))] \tag{5}$$

其中 G 是生成器,D 是判别器,E 是求期望,x 是真实数据样本,z 是噪声样本,$G(z)$ 是生成器生成的数据.

GAN 自提出以来,在各种无监督和半监督学习任务中得到广泛应用[7]. 但由于原始 GAN 的学习模型过于自由而导致训练过程和结果都不可控,经常出现梯度消失或模式坍塌等问题. 为解决上述问题,有学者提出 WGAN-GP 模型[12],利用 Wasserstein 距离代替 JS 散度来更准确地度量生成分布与真实分布间的差异程度,同时施加梯度惩罚,进一步解决训练稳定性的问题. 生成器和判别器的损失函数分别如式(6)和式(7)所示.

$$\arg\min_w -D_w(G_\theta(z)) \tag{6}$$

$$\arg\min D_w(G_\theta(z))-D_w(x)+\lambda(\|\nabla_{\hat{x}}D_w(\hat{x})\|_2-1)^2 \tag{7}$$

其中 $\hat{x}=\varepsilon x+(1-\varepsilon)G_\theta(z)$,$\varepsilon$ 是服从[0,1]均匀分布的随机数. 该梯度惩罚作为函数的正则化项能够加快 WGAN-GP 的收敛速度.

2.3　差分隐私 GAN 算法

差分隐私 GAN 是指用差分隐私的方式训练 GAN 模型,使其合成的数据在高度近似原始数据的前提下还符合差分隐私特性,可满足多种数据分析任务. Xie 等[8]提出差分隐私 GAN 的通用实现方法 DPGAN,其以 WGAN-GP 为基本框架,算法流程可总结为"梯度裁剪加噪声→更新判别器→更新生成器",具体地:在判别器的每次更新中,先从原始数据集中抽样并计算梯度(如 step4 ~ step5),然后裁剪梯度并添加噪声(如 step6),确保敏感度以阈值 C 为界并实现差分隐私;更新判别器参数后(如 step7),从噪声分布 p_z 中抽样并更新生成器参数(如 step9 ~ step10);同时,利用 Abadi 等[13]提出的隐私统计计算训练过程中的隐私损失(如 step11). 算法以对抗学习的形式循环迭代(step2 ~ step13),直到累积隐私损失超过总隐私预算(如 step12)或迭代结束时算法终止. 其伪代码[8]如算法 1 所示.

算法 1　DPGAN 算法

输入:源数据集 $X=(x_1,x_2,\cdots,x_{|D|})$,批训练数据大小 m,生成器迭代次数 T_g,生成器每迭代 1 次时判别器的迭代次数 T_d,梯度惩罚系数 λ,梯度裁剪阈值 C,噪声规模 σ,Adam 超参数(α,β_1,β_2),总隐私预算(ε_0,δ_0).

输出:差分隐私生成器 G

1)初始化生成器参数 θ 和判别器参数 w
2) for $t_1\in range(0,T_g)$
3)　for $t_2\in range(0,T_d)$
4)　　sample $\{x_i\}_{i=1}^m\sim p_{\text{data}}$　//从源数据集分布 p_{data} 中抽样
5)　　$\{g^{(i)}\}_{i=1}^m\leftarrow Gradient(\{x_i\}_{i=1}^m,m)$　//计算梯度
6)　　$g^{(i)}\leftarrow g^{(i)}/\max(1,\|g^{(i)}\|_2/C)+N(0,(\sigma C)^2\boldsymbol{I}),1\leqslant i\leqslant m$　//梯度裁剪并添加噪声
7)　　$w\leftarrow Adam\left(\frac{1}{m}\sum_{i=1}^m g^{(i)},w,\alpha,\beta_1,\beta_2\right)$　//更新判别器参数
8)　end for
9)　从噪声 p_z 中抽样 $\{z^{(i)}\}_{i=1}^m\sim p_z$
10)　$\theta\leftarrow Adam(\nabla_\theta\frac{1}{m}\sum_{i=1}^m -D_w(G_\theta(z^{(i)})),\theta,\alpha,\beta_1,\beta_2)$　//更新生成器参数
11)　计算算法消耗的隐私预算 ε(计算过程见 3.5 节中的时刻统计 $\alpha(\lambda)$ 和定理 1)
12)　若 $\varepsilon>\varepsilon_0$,则结束循环
13) end for
14)返回生成器 G

Procedure Gradient($\{x_i\}_{i=1}^m$,m)

1) for $i=1,2,\cdots,m$ do
2)　sample $z\sim p_z$,$\varepsilon\sim U(0,1)$
3)　$\hat{x}=\varepsilon x_i+(1-\varepsilon)G_\theta(z)$
4)　$l^{(i)}\leftarrow D_w(G_\theta(z))-D_w(x_i)+\lambda(\|\nabla_{\hat{x}}D_w(\hat{x})\|_2-1)^2$
5)　$g^{(i)}\leftarrow\nabla_w l^{(i)}$
6) end for
7) return $\{g^{(i)}\}_{i=1}^m$

3　优化策略

GAN 自发布以来由于其不稳定性而备受诟病,在与差分隐私机制的结合中又添加额外的噪声,导致其不稳定性问题更加突出. 算法 1 在判别器上添加固定大小的噪声并进行统一梯度裁剪,导致模型出现以下问题:①合成数据可用性低,难以用于其它数据分析任务;②相比常规 GAN 模型收敛速度慢,导致隐私损失过大.

梯度裁剪和噪声添加是差分隐私 GAN 的两大核心步骤. 本文从这两个核心步骤着手,设计动态隐私预算分配(DPBA,Dynamic Privacy Budget Allocation)、自适应

裁剪阈值选取(ACTS,Adaptive Clipping Threshold Selection)、权重参数聚类(WC,Weight Parameters Clustering)等多种优化策略来改进 DPGAN 算法,在合理的隐私预算下提高生成数据的质量和模型收敛速度. 下面着重介绍这三种优化策略.

3.1 动态隐私预算分配

隐私预算 ε 代表着数据隐私保护程度,ε 值越大表示隐私保护程度越低. 由定义 3 和算法 1 中的 step6 可知,分配给每次迭代的隐私预算决定了该次迭代中添加的高斯噪声大小 σ. 算法 1 在每次迭代中添加相同规模的高斯噪声,这会导致模型在训练后期由于噪声较大而难以收敛. 因此,对于总隐私预算固定的差分隐私 GAN 算法来说,迭代过程中如何分配隐私预算将决定生成数据的质量.

针对动态隐私预算分配问题,傅继彬等[4]根据决策树各层次间的分支关系,利用类几何分配机制来决定每一层分配的隐私预算;Wang 等[14]根据社交网络实时数据的变化趋势自适应分配隐私预算;李万杰等[15]根据用户等级和数据属性的重要程度设置不同的隐私预算,从而在合并数据集时加入对应的噪声. 但是,上述方法均只适用于结构化数据集,且不能应用于以多次迭代为特点的深度学习算法. 由于差分隐私 GAN 算法本身计算量较大,因此需要设计一种计算开销较小的隐私预算分配策略,同时应便于追踪隐私损失. 文献[16]在训练过程中使用递减的学习速率,取得了比固定学习速率的深度学习算法更高的准确性. 鉴于这一思想,本文提出如下动态隐私预算分配策略:随着模型的收敛逐渐增加分配给每次迭代的隐私预算,即逐渐减小添加在梯度上的噪声大小 σ,使模型在训练后期更容易接近局部最优从而取得更高的准确性.

为了不造成额外的隐私损失,本文在不接触原始隐私数据的前提下设计三种可选的噪声衰减函数,使噪声规模 σ 以训练时期(epoch)为单位周期性地更新(在一个 epoch 内的所有迭代 iteration 中噪声依然保持不变). 具体如下:

(1)指数式噪声衰减

$$\sigma_t = \sigma_0 \mathrm{e}^{-kt} \tag{8}$$

其中 σ_0 是初始噪声大小,t 是当前时期数,k ($k>0$)是衰减率.

(2)周期式噪声衰减

$$\sigma_t = \sigma_0 * k^{\lfloor t/period \rfloor} \tag{9}$$

其中参数 *period* 是噪声的更新周期.

(3)多项式噪声衰减

$$\sigma_t = (\sigma_0 - \sigma_{\mathrm{end}}) * (1 - t/period)^k + \sigma_{\mathrm{end}} \tag{10}$$

其中 σ_{end}($\sigma_{\mathrm{end}} < \sigma_0$)是噪声的最终大小. 当训练时期 $t <$ *period* 时,噪声根据式(10)不断减小至 σ_{end};当 $t >$ *period* 时,噪声保持 σ_{end}不变.

以上三种噪声衰减函数计算简单,在算法训练过程中可灵活调整实施策略:比如在训练初期为了加快收敛使用固定噪声大小,当训练进入中后期时才使用该衰减策略;在指数式衰减函数中用$\lfloor t/period \rfloor$代替 t,加大噪声更新周期来稳定训练效果.

噪声衰减函数中的衰减率 k 和初始噪声大小 σ_0 等超参数对于模型精度有重要的影响. 为了选取最优的超参数,一种可选的方法是训练 m 个生成对抗网络来测试 m 种超参数取值方案,最终选取生成数据质量最高的取值方案. 但是该方法会使隐私预算增加 m 倍,隐私损失较大. 因此本文采用一种满足差分隐私特性的选值方案:将训练数据集均分为 $m+1$ 份,其中 m 份数据分别用于训练 m 种超参数取值下的 DPGAN 模型并用其生成数据训练 CNN 分类器,余下的 1 份数据用于测试这 m 个分类器的准确率. 记 z_i($1 \leqslant i \leqslant m$)为第 i 个分类器在测试集上错误分类的个数,则以正比于 $\exp\left(\frac{-\varepsilon z_i}{2}\right)$的概率选取第 i 种超参数取值方案. 根据定义 4 中的指数机制,该选值方案满足 ε-差分隐私.

3.2 自适应裁剪阈值选取

在算法 1[8]和 Abadi 等[13]提出的差分隐私随机梯度下降算法中,梯度裁剪阈值 C 均是作为输入且并未说明如何计算. 实际上,C 值对于深度学习隐私保护算法的影响很大:由算法 1 中的 step6 可知,C 值太小会使梯度过度裁剪而造成模型收敛缓慢,C 值过大会添加过量噪声而导致生成数据质量低. 理想情况下,对于每个样本 x_i 令 $C_i \approx \| g(x_i) \|_2$ 是最优的,此时每个样本都能以最小误差对平均梯度做出最大贡献. 但是样本的梯度在迭代过程中不断变化,因此很难找到全局最优的固定裁剪阈值 C.

为此,有学者[10,11]提出自适应裁剪阈值选取方案:在每次迭代中从少量公开数据 D_{pub} 中采样并计算梯度均值作为裁剪阈值,但是公开数据集在很多情况下难以获取. 还有一种简单的取值方案:每次迭代中取加噪后的梯度均值作为裁剪阈值,即 $C_s = \frac{1}{|L|}\sum_i \| g(x_i) \|_2 + N\left(0, s \cdot \frac{\sigma'}{L}\right)$,但当存在异常样本 x_i 时,$s \geqslant \max \| g(x_i) \|_2$ 这一约束会导致 s 值过大而添加过量噪声.

针对上述问题,本文提出一种基于高斯机制的自适应裁剪阈值选取方法,如算法 2:令 $C_{\max}$ 为当前批次中最大的梯度范数,即 $C_{\max} = \max\{ \| g(x_i) \|_2, 1 \leqslant i \leqslant L \}$. 将 $(0, C_{\max})$ 均分为 r 个小区间 $(C_{i-1}, C_i]$(如 step1),记每个区间 $(C_{i-1}, C_i]$ 的频数 t_i 为梯度范数在该区间内的样本数(如 step2). 将所有区间的频数添加高斯噪声 $\xi \sim N(0, (\sqrt{2}\sigma_C)^2 \boldsymbol{I})$ 后(如 step3),选取最大值所对应的

区间上界作为该轮迭代的梯度裁剪阈值.

由算法 2 可知:①即使出现某个异常样本 x_i 使 $\max \| g(x_i) \|_2$ 过大,也最多只能改变两个区间的频数,所以算法 2 的 L_2-敏感度为 $\Delta f = \sqrt{2}$. 根据定义 3,通过添加适量噪声 $\xi \sim N(0, (\sqrt{2}\sigma_C)^2 \boldsymbol{I})$,可确保算法 2 满足差分隐私特性;②裁剪阈值 C 接近于当前批次中大部分样本的梯度范数,即 $C \approx (1/L) \sum_i \| g(x_i) \|_2$,因此该取值是近似最优的,可加快算法的收敛速度.

算法 2　自适应裁剪阈值选取(ACTS)

输入:当前批次样本的梯度 $S = (g(x_1), g(x_2), \cdots, g(x_L))$,噪声规模 σ_C,区间数 r

输出:裁剪阈值 C_s

1) $C_j \leftarrow j \cdot C_{\max}/r$ for $0 \leqslant j \leqslant r$　//离散化区间$(0, C_{\max})$

2) $t_j = |\{ g(x_i) \in S : C_{j-1} < \| g(x_i) \|_2 \leqslant C_j \}|, 0 \leqslant j \leqslant r$　//频数 t_j 等于梯度范数落在该区间内的样本数

3) $C_s \leftarrow C_j$ where $t_j = \arg\max_{j \geqslant 1} \{ t_j + N(0, (\sqrt{2}\sigma_C)^2 \boldsymbol{I}) \}$　//选择加噪后最大频数对应的区间上界

4)返回裁剪阈值 C_s

3.3　权重参数聚类

算法 1 及文献[10,13]中所有参数的梯度被统一裁剪(如 step6),该方法最小化了每次迭代消耗的隐私预算,但实验中发现:GAN 模型中大部分偏置参数的梯度接近于 0,而权重参数的梯度远大于 0 且彼此相差较大. 这就造成一个矛盾:若采用统一的裁剪方法,部分权重参数的梯度会被添加过量噪声,进而导致模型收敛缓慢;若针对每一个参数都单独进行梯度裁剪,虽能减小总体噪声规模,却会消耗较多的隐私预算. 为此,本文提出权重参数聚类策略:①所有偏置参数依然进行统一的梯度裁剪;②采用密度聚类算法 DBSCAN 将权重参数分成不同的类,并针对每类参数进行分类梯度裁剪.

对于策略①,由于偏置参数数量多且梯度接近于 0,统一的梯度裁剪消耗的隐私预算较小且基本不影响模型收敛速度;对于策略②,基于算法 2 计算出的每个权重参数 w_i 的裁剪阈值 $c(w_i)$,将参数聚类并针对每类参数进行更精确的分类梯度裁剪,可减小不必要的噪声进而提高模型的精度. 由于密度聚类算法 DBSCAN 不需预先指定类的个数且计算复杂度小,故采用该算法对权重参数进行聚类.

具体过程如算法 3 所示:首先将所有权重参数及其对应的裁剪阈值组成集合 G(如 step1),然后以每个未聚类的参数 w_i 为中心点,将拥有相近裁剪阈值且区域密度达到一定要求(即裁剪阈值相差不超过 μ 的参数个数不少于 *Minpts*,如 step3 ~ step4)的所有参数聚为一类,同时合并裁剪阈值(如 step11). 循环迭代,直到所有参数都被聚为某一类中.

算法 3　权重参数聚类(WC)

输入:所有权重参数 w_i 及其对应的裁剪阈值 $c(w_i)$,邻域半径 μ,最小邻域点数 *Minpts*

输出:类的个数 n,每个类中参数集合及对应的裁剪阈值

1) $G \leftarrow \{ (w_i : c(w_i)) \}_i$,标记所有的参数 w_i 为未访问,$n = 1$

2) for each $w_i \in G$ 且未访问

3)　标记 w_i 已访问,$NEps(w_i) = \{ w_j \in G : |c(w_j) - c(w_i)| \leqslant \mu \}$
　//$NEps(w_i)$ 为 w_i 的 μ 邻域参数集合

4)　if $|NEps(w_i)| \geqslant Minpts$

5)　　创建新类 $G_n = \{ w_i \}$,令类的裁剪阈值为 $c(G_n) = c(w_i)$

6)　　for each $w_j \in NEps(w_i)$ and w_j 未访问

7)　　标记 w_j 已访问,$NEps(w_j) = \{ w_l \in G : |c(w_l) - c(w_j)| \leqslant \mu \}$

8)　　if $|NEps(w_j)| \geqslant Minpts$

9)　　$NEps(w_i) \leftarrow NEps(w_i) + NEps(w_j)$　//将 w_j 的 μ 邻域合并入 w_i 的 μ 邻域中

10)　　if w_j 未被加入类

11)　　$G_n \leftarrow G_n + \{ w_j \}$, $c(G_n) = \sqrt{c(G_n)^2 + c(w_j)^2}$　//聚类并更新裁剪阈值

12)　end for

13)　n ++

14) end for

3.4　算法描述

结合三种优化策略,本文提出改进后的差分隐私 GAN 算法(IDPGAN,Improved Differential Privacy Generative Adversarial Network). 具体过程如算法 4 所示. 相比算法 1,算法 4 的总体流程"梯度裁剪加噪声→更新判别器→更新生成器"保持不变,主要实现以下三点改进:第一,引入动态隐私预算分配策略,根据训练进度逐渐减小噪声规模,如 step3;第二,不再使用固定的梯度裁剪阈值 C,而在每轮迭代中自适应地选取阈值,如 step7;第三,将权重参数聚成 n 个类 $\{ (G_j, c_j) \}_{j=1}^n$,每类参数共享同一裁剪阈值 c_j(如 step8),进行梯度分类裁剪并添加动态噪声(如 Procedure cluster-clipping). 其余操作同算法 1 保持一致.

算法 4　改进后的差分隐私 GAN(IDPGAN)

输入:源数据集 $X = (x_1, x_2, \cdots, x_{|D|})$,批训练数据大小 m,判别器中权重参数的个数 n_{param},生成器迭代次数 T_g,生成器每迭代 1 次时判别器的迭代次数 T_d,梯度惩罚系数 λ,噪声规模 σ_c,σ_0,衰减率 k,区间数 r,*Adam* 超参数$(\alpha, \beta_1, \beta_2)$,邻域半径 μ,最小邻域点数 *Minpts*,总隐私预算$(\varepsilon_0, \delta_0)$.

输出:差分隐私生成器 G

1)初始化生成器参数 θ 和判别器参数 w

2) for $t_1 \in range(0, T_g)$

3)　$t = 0.01 t_1$, $\sigma_t = DPBA(\sigma_0, k, t)$　//优化策略 1,更新噪声规模

4)　for $t_2 \in range(0, T_d)$

5) sample $\{x_i\}_{i=1}^{m} \sim p_{\text{data}}$ //从源数据集分布 p_{data} 中抽样

6) $\{g^{(i)}\}_{i=1}^{m} \leftarrow Gradient(\{x_i\}_{i=1}^{m}, m)$ //*Gradient* 同算法 1 中的一致

7) $\{c(w_i)\}_{i=1}^{n_{\text{param}}} \leftarrow ACTS(\{g^{(i)}\}_{i=1}^{m}, \sigma_c)$ //优化策略 2,确定每个参数的裁剪阈值

8) $\{(G_j, c_j)\}_{j=1}^{n} \leftarrow WC(\{w_i, c(w_i)\}_{i=1}^{n_{\text{param}}}, \mu, Minpts)$ //优化策略 3,将权重参数聚成 n 类

9) $\{g_j^{(i)}\}_{i=1,j=1}^{m,n} \leftarrow cluster-clipping(\{(G_j, c_j)\}_{j=1}^{n}, \{g^{(i)}\}_{i=1}^{m})$ //梯度分类裁剪并添加噪声

10) $w_j \leftarrow Adam\left(\frac{1}{m}\sum_{i=1}^{m} g_j^{(i)}, w_j, \alpha, \beta_1, \beta_2\right)$ for $j=1,2,\cdots,n$ //更新判别器参数

11) end for

12) sample $\{z^{(i)}\}_{i=1}^{m} \sim p_z$

13) $\theta \leftarrow Adam(\nabla_\theta \frac{1}{m}\sum_{i=1}^{m} -D_w(G_\theta(z^{(i)})), \theta, \alpha, \beta_1, \beta_2)$ //更新生成器参数

14) 计算算法消耗的隐私预算 ε(计算过程见 3.5 节中的时刻统计 $\alpha(\lambda)$ 和定理 2)

15) 若 $\varepsilon > \varepsilon_0$,则结束循环

16) end for

17) 返回生成器 G

Procedurecluster-clipping($\{(G_j, c_j)\}_{j=1}^{n}, \{g^{(i)}\}_{i=1}^{m}$)

1) for $i=1,2,\cdots,m$ do

2) $g_j^{(i)} \leftarrow g^{(i)} \cap G_j$ for $j=1,2,\cdots,n$

3) for $j=1,2,\cdots,n$ do

4) $\xi \sim N(0, (\sigma_t c_j)^2 \boldsymbol{I})$

5) $g_j^{(i)} \leftarrow \frac{g_j^{(i)}}{\max(1, \|g_j^{(i)}\|_2 / c_j)} + \xi$ //分类裁剪并添加噪声

6) end for

7) end for

8) return $\{g_j^{(i)}\}_{i=1,j=1}^{m,n}$

3.5 算法隐私性分析

如何跟踪算法训练过程中的隐私损失是差分隐私深度学习算法的关键部分之一. 本文采用 Abadi 等[13]提出的时刻统计来计算累积隐私损失. 首先给出以下定义:

定义 5[13] (隐私损失). 令 $A: D \rightarrow R$ 为随机算法,D 和 D' 为最多相差一条记录的相邻数据集,A 在输出 $O \in R$ 处的隐私损失为随机变量

$$c(O, A, D, D') \triangleq \log \frac{\Pr[A(D)=O]}{\Pr[A(D')=O]} \tag{11}$$

其中概率 $\Pr[\cdot]$ 由算法 A 的随机性决定.

定义 6[13] (时刻统计). 令 $A: D \rightarrow R$ 为随机算法,D 和 D' 为最多相差一条记录的相邻数据集,A 在 λ 时刻的时刻统计为

$$\alpha(\lambda) \triangleq \max_{D,D'} \log E_{O\sim A(D)}[\exp(\lambda c(O, A, D, D'))] \tag{12}$$

定理 1[13] 若算法 $A_{1:k}$ 由相互独立的子算法 $A_1, A_2, \cdots, A_k$ 组成,则有

(1)(组合性)总的时刻统计由每次迭代的时刻统计组成:$\alpha_{A_{1:k}}(\lambda) \leqslant \sum_{i=1}^{k} \alpha_{A_i}(\lambda)$.

(2)(尾界限)对于任意 $\varepsilon > 0$,算法 $A_{1:k}$ 满足(ε, $\min_\lambda \exp(\sum_{i=1}^{k} \alpha_{A_i}(\lambda) - \lambda\varepsilon)$) - 差分隐私.

由定理 1 可知,算法的隐私损失正比于迭代次数. 对于本文提出的算法 4,设 IDPGAN 中判别器的迭代次数为 T 次,总的时刻统计为 $\alpha(\lambda)$,第 t 次迭代的时刻统计为 $\alpha^t(\lambda)$. 则根据定理 1 中的组合性,由于不同迭代次数中的高斯扰动算法互相独立,有

$$\alpha(\lambda) \leqslant \sum_{t=1}^{T} \alpha^t(\lambda) \tag{13}$$

而 IPDGAN 在每一次迭代中使用了两个高斯扰动算法:①算法 a,基于当前批次的样本梯度 $\{g^{(i)}\}_{i=1}^{m}$,优化策略 ACTS 通过添加噪声 $\xi_1 \sim N(0, (\sqrt{2}\sigma_C)^2 \boldsymbol{I})$ 选取自适应裁剪阈值(如算法 4 中的 step7),令其时刻统计为 $\alpha_{\text{ACTS}}^t(\lambda)$;②算法 b,基于同一批次的样本梯度 $\{g^{(i)}\}_{i=1}^{m}$,将梯度分类裁剪并添加噪声 $\xi_2 \sim N(0, (\sigma_t c_j)^2 \boldsymbol{I})$(如 Procedure cluster-clipping),令其时刻统计为 $\alpha_{\text{cclip}}^t(\lambda)$. 虽然同一次迭代中的两个高斯算法使用了相互独立的高斯噪声,但它们均是针对于同一数据批次(见算法 4 的 step5 ~ step9),因此并不满足定理 1 中子算法相互独立的条件. 因此第 t 次迭代的时刻统计 $\alpha^t(\lambda)$ 不能简单地计算为

$$\alpha^t(\lambda) \leqslant \alpha_{\text{ACTS}}^t(\lambda) + \alpha_{\text{cclip}}^t(\lambda) \tag{14}$$

针对这种情况,文献[9]在定理 1 的基础上提出了一种通用的时刻统计,适用于互不独立的子算法.

定理 2[9] 若算法 $A_{1:k}$ 由子算法 $A_1, A_2, \cdots, A_k$ 组成,则有:

(1)(组合性)总的时刻统计可计算为 $\alpha_{A_{1:k}}(\lambda) \leqslant \sum_{i=1}^{k} j_i \alpha_{A_i}(\lambda / j_i)$.

(2)(尾界限)对于任意 $\varepsilon > 0$,算法 $A_{1:k}$ 满足(ε, $\min_\lambda \exp(\sum_{i=1}^{k} j_i \cdot \alpha_{A_i}(\lambda / j_i) - \lambda\varepsilon)$) - 差分隐私.

其中 $j_i > 0$,且 $\sum_{i=1}^{k} j_i = 1$,子算法 $A_1, A_2, \cdots, A_k$ 可互不独立. 证明过程见文献[9],本文不再赘述.

基于定理 2 的组合性,IDPGAN 的第 t 次迭代可视为由不独立的子算法 a 和子算法 b 组成,则第 t 次迭代的时刻统计 $\alpha^t(\lambda)$ 为

$$\alpha^t(\lambda) \leqslant \min_{j_1, j_2 \in (0,1): j_1 + j_2 = 1} (j_1 \alpha_{\text{ACTS}}^t(\lambda / j_1) + j_2 \alpha_{\text{cclip}}^t(\lambda / j_2)) \tag{15}$$

结合式(13)和式(15),可得 IDPGAN 算法总的时刻统计为

$$\alpha(\lambda) \leqslant \sum_{t=1}^{T} \alpha^t(\lambda)$$

$$\leqslant \sum_{t=1}^{T} \min_{j_1,j_2\in(0,1):j_1+j_2=1} (j_1\alpha_{\mathrm{ACTS}}^{t}(\lambda/j_1) + j_2\alpha_{\mathrm{cclip}}^{t}(\lambda/j_2)) \tag{16}$$

对于 $\alpha_{\mathrm{ACTS}}^{t}(\lambda/j_1)$ 和 $\alpha_{\mathrm{cclip}}^{t}(\lambda/j_2)$,为便于说明,以 λ' 统一表示 λ/j_1 和 λ/j_2,其计算过程如下[13]:

令

$$\mu_0(x|\sigma)=g(x|\sigma),\mu_1(x|\sigma)=(1-q)g(x|\sigma)+qg(1-x|\sigma)$$

其中 q 是抽样概率,在算法 4 中 $q=m/|D|$;$g(x|\sigma)=\frac{1}{\sqrt{2\pi\sigma^2}}\mathrm{e}^{-x^2/2\sigma^2}$. 则有

$$\alpha_{\mathrm{ACTS}}^{t}(\lambda')=\log\max(E_1(\lambda',\sigma_C),E_2(\lambda',\sigma_C)),$$

$$\alpha_{\mathrm{cclip}}^{t}(\lambda')=\log\max(E_1(\lambda',\sigma_t),E_2(\lambda',\sigma_t))$$

$$\text{where } E_1(\lambda',\sigma)=\int_{-\infty}^{\infty}\mu_0(x|\sigma)\cdot\left(\frac{\mu_0(x|\sigma)}{\mu_1(x|\sigma)}\right)^{\lambda'}\mathrm{d}x,$$

$$E_2(\lambda',\sigma)=\int_{-\infty}^{\infty}\mu_1(x|\sigma)\cdot\left(\frac{\mu_1(x|\sigma)}{\mu_0(x|\sigma)}\right)^{\lambda'}\mathrm{d}x \tag{17}$$

其中 σ_C 为 ACTS 策略中的噪声规模,为定量;σ_t 为第 t 次迭代时的噪声规模 $\sigma_t=\mathrm{DPBA}(\sigma_0,k,t)$,为变量.

在计算得到 IDPGAN 总的隐私统计 $\alpha(\lambda)$ 后,可得:

定理 3　对于任意的 δ,IDPGAN 算法满足 $(\min_\lambda(\alpha(\lambda)-\log\delta)/\lambda,\delta)$-差分隐私.

证明　根据定理 1 中的尾界限特性,IDPGAN 算法满足 $(\varepsilon,\min_\lambda\exp(\alpha(\lambda)-\lambda\varepsilon))$-差分隐私. 而 IDPGAN 是通过追踪训练过程中消耗的隐私预算 ε 来判断算法是否停止迭代(如算法 4 中的 step14 ~ step15),因此需要根据时刻统计 $\alpha(\lambda)$ 计算 ε. 已知 $\delta=\min_\lambda\exp(\alpha(\lambda)-\lambda\varepsilon)$,可推导得到 $\varepsilon=\min_\lambda(\alpha(\lambda)-\log\delta)/\lambda$.

因此 $(\varepsilon,\min_\lambda\exp(\alpha(\lambda)-\lambda\varepsilon))$-差分隐私的等价形式是 $(\min_\lambda(\alpha(\lambda)-\log\delta)/\lambda,\delta)$-差分隐私.

算法在实际运行中,整数 λ 的取值范围通常为 $0\leqslant\lambda\leqslant100$,$\delta=1/|D|$,对于式(15)中的 j_1 和 j_2,考虑 10 个不同的取值,以得到足够准确的 $\alpha(\lambda)$,进而可以精确地计算出 IDPGAN 满足 $(\min_\lambda(\alpha(\lambda)-\log\delta)/\lambda,\delta)$-差分隐私.

4　实验

本实验在结构化数据集 Poker Hand 和非结构化数据集 MNIST、CelebA 上验证 IDPGAN 算法生成数据的质量、可用性、训练效率及优化策略的有效性.

4.1　实验设置

本实验使用 3 种公开数据集.

(1) MNIST:包含 70000 张 10 类手写数字图片,每张图片大小为 28 × 28,训练集中有 60000 张图片和标签,测试集中有 10000 张图片和标签.

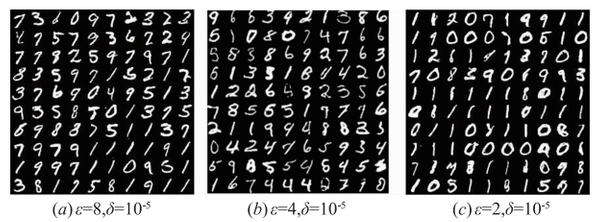
(2) CelebA:包含 200000 张名人的图片,每张图片大小为 48 × 48 且被标记 40 种特征属性.

(3) Poker Hand:UCI 上的公开数据集,训练集有 25010 个样本,测试集有 1000000 个样本,每个样本包含 5 张扑克牌的"大小""花色"以及由它们共同决定的"牌型"等 11 个属性. 其中"大小"有 13 种,花色有 4 种,牌型有 10 种,包括"一对""三带二""同花顺"等,是样本的标签.

实验中 IDPGAN 采用文献[12]中的网络结构来构建判别器和生成器,相关参数设置与文献[12]相同,即判别器和生成器的迭代次数分别为 $T_d=5$,$T_g=5\times10^5$,梯度惩罚项 $\lambda=10$,数据批次大小为 $m=64$,Adam 超参数 $\alpha=0.0001$,$\beta_1=0$,$\beta_2=0.9$. 对于 DPBA 策略中的初始噪声大小 σ_0 和衰减率 k,根据 3.1 节中的差分隐私选值方案,以 $\sigma_0\in[7,15]$,$k\in[0.005,0.04]$ 为测试范围,最终选取最优值 $\sigma_0=10$,$k=0.02$. 对于 ACTS 策略中的噪声规模 σ_C 和区间数 r,以 $\sigma_C\in[2,6]$,$r\in[80,140]$ 为测试范围,发现不同取值对于实验结果影响很小,本实验中取 $\sigma_C=4$,$r=100$. 对于 WC 策略中的邻域半径 μ 和最小邻域点数 *Minpts*,根据实验中裁剪阈值的分布以及 IDPGAN 权重参数的个数,以 $\mu\in[1,4]$,$Minpts\in[1500,4000]$ 为测试范围,最终选取最优值 $\mu=3$,$Minpts=2000$. 生成器中噪声 z 的维度为 100,每维都限制在 $[-1,1]$ 范围内. 默认采用 3.1 节中的指数式噪声衰减来实现 IDPGAN.

4.2　生成数据质量对比

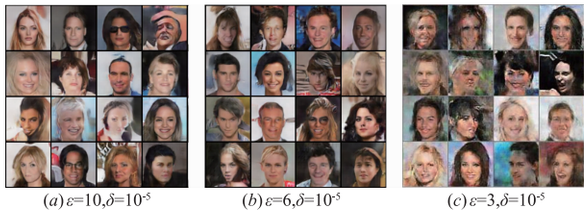
(1)对于非结构化的图像数据集,首先从视觉效果上展示 IDPGAN 在不同隐私预算 ε 下生成图像的质量. 由图 1 和图 2 可以看出,IDPGAN 生成的图像不仅十分逼近真实图像,而且还具有自己独特的细节. 隐私预算 ε 值越大(即添加噪声越小),生成的图像越清晰,但也意味着隐私保护度有所减弱. 因此,选取合适的 ε 值可在数据可用性和隐私性间取得平衡. 同时,由图 1 和图 2 的合成样本可知,任何得到合成样本的用户也难以推测某个样本是否在训练数据集中,因此保护了原始训练数据集的隐私.



(a) $\varepsilon=8,\delta=10^{-5}$　(b) $\varepsilon=4,\delta=10^{-5}$　(c) $\varepsilon=2,\delta=10^{-5}$

图1　MNIST上不同隐私预算下的合成图像

(2)对于有标签数据集 MNIST 和 Poker Hand,使用常用标准 Inception Scores[12] 来评价 IDPGAN 生成数据



(a) ε=10,δ=10^{-5} (b) ε=6,δ=10^{-5} (c) ε=3,δ=10^{-5}

图2 CelebA上不同隐私预算下的合成图像

的质量,其定义如下:

$$s(G)=\exp(E_{x\sim G(z)}KL(\Pr(y|x)\parallel\Pr(y))) \quad (18)$$

其中 x 是生成器 G 生成的样本,$\Pr(y|x)$ 是样本 x 属于类别 y 的条件概率分布,若 x 十分逼近真实样本(即向量 y 的某一维度值格外大),则该分布呈尖锐状;$\Pr(y)=\int\Pr(y|x=G(z)\mathrm{d}z)$ 是全体生成样本在所有类别上的边缘分布,若生成样本的多样性好,则该分布呈均匀分布. 因此,$s(G)$越大(即这 2 项分布的 KL 距离越大),说明生成样本的质量和多样性越好.

对比算法选择 WGAN-GP[12]、DPGAN[8]、GANobfuscato[10]、DPACGAN[11],同时令 IDPGAN 取不同的隐私预算,实验结果如表 1 所示(real data 代表真实数据).

由表 1 可知:①当隐私预算分别在 MNIST 上取 ε=4、Poker Hand 上取 ε=15 时,GANobfuscator、DPACGAN 和 IDPGAN 的 score 值均大于 DPGAN,原因在于前三者算法均采用了自适应方法来选取梯度裁剪阈值,有效减小了不必要的噪声;其中 IDPGAN 的 score 值最大,说明 IDPGAN 通过有效组合三种优化策略提高了生成数据的质量. ②对比 IDPGAN 在不同隐私预算下的 score,发现在 MNIST 上取 ε=4、Poker Hand 上取 ε=15,可较好地平衡生成数据的质量与隐私性.

(3)对于无标签数据集 CelebA,为了评估 IDPGAN 生成数据的质量,令其总隐私预算分别为 ε=3,6,10,依然采用上述四种算法做对比. 用真实数据训练另一个判别器 D',并测试 D'能否将真实数据与生成数据区分开来. 令 $\Pr(y|x)$表示 D'判断样本 x 来源(真实数据或生成数据)的概率分布,B_p表示 p=0.5 的伯努利分布,这两项分布的 JS 散度可有效衡量生成数据与真实数据的相似程度,如式(19). $s'(G)$越小,说明 D'越难分辨生成数据和真实数据,则代表生成数据质量越高.

$$s'(G)=\frac{1}{2}KL(\Pr(y|x)\parallel B_p)+\frac{1}{2}KL(B_p\parallel\Pr(y|x)) \quad (19)$$

由表2 可知:①当 IDPGAN 的隐私预算分别取 ε=3 和 ε=6 时,其 JS 散度均较大,说明对于纹理较为复杂的图像数据集 CelebA,隐私预算过小会严重降低生成数据的质量. ②当隐私预算均为 ε=10 时,相比于 DPGAN、GANobfuscator 和 DPACGAN,IDPGAN 的 JS 散度值最小,且与无隐私保护的 WGAN-GP 相差不大,因此 IDPGAN 在 CelebA 数据集上令隐私预算为 10.

表 1 真实数据和合成数据的 Inception scores

Dataset	Alg	$n(\times10^4)$	(ε,δ)	Score
MNIST	real data	6	—	9.96 ±0.03
	WGAN-GP	6	—	9.21 ±0.03
	DPGAN	6	$(4,10^{-5})$	8.64 ±0.03
	GANobfuscator	6	$(4,10^{-5})$	8.89 ±0.02
	DPACGAN	6	$(4,10^{-5})$	8.81 ±0.03
	IDPGAN	6	$(2,10^{-5})$	8.56 ±0.02
	IDPGAN	**6**	$\mathbf{(4,10^{-5})}$	**8.93 ±0.03**
	IDPGAN	6	$(8,10^{-5})$	8.96 ±0.03
Poker Hand	real data	2.5	—	4.43 ±0.01
	WGAN-GP	2.5	—	3.16 ±0.01
	DPGAN	2.5	$(15,10^{-5})$	2.23 ±0.01
	GANobfuscator	2.5	$(15,10^{-5})$	2.54 ±0.01
	DPACGAN	2.5	$(15,10^{-5})$	2.39 ±0.01
	IDPGAN	2.5	$(10,10^{-5})$	2.25 ±0.01
	IDPGAN	**2.5**	$\mathbf{(15,10^{-5})}$	**2.65 ±0.01**
	IDPGAN	2.5	$(20,10^{-5})$	2.78 ±0.02

表 2 真实数据和合成数据的 JS 散度

Dataset	Alg	$n(\times10^5)$	(ε,δ)	JS 散度
CelebA	real data	2	—	0
	WGAN-GP	2	—	0.09 ±0.01
	DPGAN	2	$(10,10^{-5})$	0.29 ±0.01
	GANobfuscator	2	$(10,10^{-5})$	0.26 ±0.01
	DPACGAN	2	$(10,10^{-5})$	0.24 ±0.01
	IDPGAN	2	$(3,10^{-5})$	0.42 ± 0.02
	IDPGAN	2	$(6,10^{-5})$	0.30 ± 0.01
	IDPGAN	2	$(10,10^{-5})$	**0.21 ±0.01**

4.3 生成数据可用性对比

GAN 生成数据的可用性是指其能否代替真实数据以用于通用的数据分析任务. 对于数据集 MNIST 和 Poker Hand,分别用标准训练集训练 IDPGAN,然后用其生成的数据来训练 CNN 分类器(LeNet-type[17]). 训练完毕后,用标准测试集来测试 CNN 的分类准确率. 准确率越高,说明 IDPGAN 生成的数据可用性越好. 实验在不同的隐私预算下进行,并采用 4.2 节中的对比算法.

由表 3 可知:①当隐私预算分别在 MNIST 上取 ε=4 和 Poker Hand 上取 ε=15 时,在四种差分隐私 GAN 方法中,DPGAN 的分类准确率最低,说明差分隐私 GAN 若不采用任何优化策略,其生成的数据可用性较

低;IDPGAN 通过在训练过程中采用多种优化策略,其生成数据的可用性最高,仅略低于无隐私保护机制的 WGAN-GP. ②IDPGAN 在 MNIST 和 Poker Hand 数据集上分别取 $\varepsilon=4$ 和 $\varepsilon=15$ 时,能够较好地平衡生成数据的可用性与隐私性.

表 3　不同方法的生成数据在标准数据集上的分类准确率

Dataset	IDPGAN				real data	WGAN-GP	DPGAN	GANobfuscator	DPACGAN
MNIST	$\varepsilon=1$	$\varepsilon=2$	$\varepsilon=4$	$\varepsilon=8$	—	—	$\varepsilon=4$	$\varepsilon=4$	$\varepsilon=4$
	85.8%	92.7%	**96.1%**	97.3%	99.6%	97.9%	94.2%	95.4%	94.7%
Poker Hand	$\varepsilon=5$	$\varepsilon=10$	$\varepsilon=15$	$\varepsilon=20$	—	—	$\varepsilon=15$	$\varepsilon=15$	$\varepsilon=15$
	41.38%	47.56%	**51.25%**	52.51%	58.42%	54.83%	48.43%	50.07%	49.30%

4.4　训练效率对比

由 2.2 节可知,WGAN-GP 模型中 Wasserstein 距离用于度量生成分布与真实分布的差异,而 DPGAN、GANobfuscator 和 IDPGAN 都是以 WGAN-GP 为基本框架,因此通过观察这三种算法在训练过程中的 Wasserstein 距离来判断收敛情况. 它们在三个数据集上的实验结果如图 3 所示. 由图 3 可知:当三种算法的隐私预算相同时,IDPGAN 在三个数据集上收敛的 Wasserstein 距离均是最小,说明本文提出的优化策略使得算法最终生成的数据更加接近真实数据;相比于 GANobfuscator,IDPGAN 的运行时间稍长,这是由于 IDPGAN 在训练时额外实施了梯度分类裁剪. 但即使对于训练样本较大、纹理较为复杂的 CelebA 数据集,IDPGAN 的运行时间也仅比 GANobfuscator 多出约 10%,说明 IDPGAN 可应用于大型数据集.

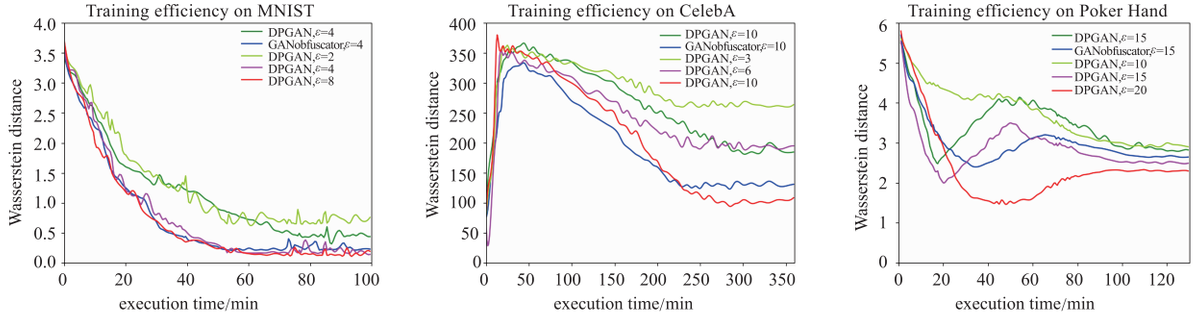


图3　训练效率对比

4.5　优化策略有效性验证

为了验证优化策略的有效性,将三种策略以不同的方式组合起来形成新的 GAN 模型,分别将 MNIST、Poker Hand 和 CelebA 作为训练数据集,计算其生成数据的 Inception score 和 JS 散度. 由表 4 可知:对于数据集 MNIST、Poker Hand 和 CelebA,当不采用任何优化策略时(即 DPGAN),生成数据的质量最差;当采用任意一种或两种优化策略时,大部分 GAN 模型生成的数据质量都会有所提高;当组合三种优化策略时(即 IDPGAN),生成数据的 Inception score 值最大,JS 散度值最小. 由此证明了三种优化策略的有效性.

表 4　不同策略组合的 Inception score 和 JS 散度

组合策略	DPBA	ACTS	WC	MNIST:Inception Score	Poker Hand:Inception Score	CelebA:JS 散度
1				8.64 ± 0.03	2.23 ± 0.01	0.29 ± 0.01
2	√			8.71 ± 0.04	2.34 ± 0.01	0.25 ± 0.02
3		√		8.61 ± 0.02	2.40 ± 0.01	0.29 ± 0.01
4	√	√		8.82 ± 0.03	2.58 ± 0.01	0.23 ± 0.01
5		√	√	8.78 ± 0.04	2.43 ± 0.01	0.32 ± 0.02
6	√	√	√	**8.93 ± 0.03**	**2.65 ± 0.01**	**0.21 ± 0.01**

5　结束语

针对现有隐私保护方法在消除敏感信息后出现数据可用性低的问题,本文提出基于生成对抗网络的隐私数据发布算法,可以无限量生成与源数据高度近似且满足差分隐私特性的合成数据. 设计动态隐私预算分配、自适应裁剪阈值选取、权重参数聚类三种优化策略来提高模型的训练稳定性和生成数据的质量,并利用时刻统计精确地分析了算法训练过程的隐私损失. 实验证明本方法合成的数据具有较好的隐私性和可用

性.下一步工作将研究更多类型数据的隐私发布如文本、序列数据集等.

参考文献

[1] Fredrikson M,Jha S,Ristenpart T. Model inversion attacks that exploit confidence information and basic countermeasures [A]. Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security [C]. USA:ACM,2015. 1322 – 1333.

[2] Cynthia Dwork,Aaron Roth. The Algorithmic Foundations of Differential Privacy[M]. USA:Now Foundations and Trends,2014.

[3] 杨高明,朱海明,方贤进,等. 局部差分隐私约束的关联属性不变后随机响应扰动[J]. 电子学报,2019,47(5):1079 – 1085.
YANG Gao-ming,ZHU Hai-ming,FANG Xian-jin,et al. Invariant post-random response perturbation for correlated attributes under local differential privacy constraint[J]. Acta Electronica Sinica,2019,47(5):1079 – 1085. (in Chinese)

[4] 傅继彬,张啸剑,丁丽萍. MAXGDDP:基于差分隐私的决策数据发布算法[J]. 通信学报,2018,39(3):136 – 146.
FU Ji-bin,ZHANG Xiao-jian,DING Li-ping. MAXGDDP:decision data release with differential privacy[J]. Journal of Communications,2018,39(3):136 – 146. (in Chinese)

[5] Zhang J,Cormode G,Procopiuc C M,et al. Privbayes:Private data release via Bayesian networks[J]. ACM Transactions on Database Systems (TODS),2017,42(4):1 – 41.

[6] Asghar H J,Ding M,Rakotoarivelo T,et al. Differentially private release of high-dimensional datasets using the Gaussian copula[J]. arXiv Preprint,2019, arXiv:1902.01499.

[7] GOODFELLOW I,POUGET-ABADIE J,MIRZA M,et al. Generative adversarial nets[A]. Advances in Neural Information Processing Systems[C]. USA: ACM, 2014. 2672 – 2680.

[8] Xie L,Lin K,Wang S,et al. Differentially private generative adversarial network[J]. arXiv Preprint,2018, arXiv:1802.06739.

[9] Acs G,Melis L,Castelluccia C,et al. Differentially private mixture of generative neural networks[J]. IEEE Transactions on Knowledge and Data Engineering,2018,31(6):1109 – 1121.

[10] Xu C,Ren J,Zhang D,et al. GANobfuscator:Mitigating information leakage under GAN via differential privacy[J]. IEEE Transactions on Information Forensics and Security,2019,14(9):2358 – 2371.

[11] 郭鹏,钟尚平,陈开志,等. 差分隐私 GAN 梯度裁剪阈值的自适应选取方法[J]. 网络与信息安全学报,2018,4(5):10 – 20.
GUO Peng,ZHONG Shang-ping,CHEN Kai-zhi,et al. Adaptive selection method of differential privacy GAN gradient clipping thresholds[J]. Chinese Journal of Network and Information Security,2018,4(5):10 – 20. (in Chinese)

[12] Gulrajani I,Ahmed F,Arjovsky M,et al. Improved training of Wasserstein gans[A]. Advances in Neural Information Processing Systems[C]. USA:ACM,2017. 5767 – 5777.

[13] ABADI M,CHU A,GOODFELLOW I,et al. Deep learning with differential privacy[A]. The ACM SIGSAC Conference on Computer and Communications Security [C]. USA:ACM,2016. 308 – 318.

[14] Wang Q,Zhang Y,Lu X,et al. Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy[J]. IEEE Transactions on Dependable and Secure Computing,2018,15(4):591 – 606.

[15] 李万杰,张兴,曹光辉,等. 基于差分隐私保护的数据分级融合发布机制[J]. 小型微型计算机系统,2019,40(10):2252 – 2256.
LI Wan-jie,ZHANG Xing,CAO Guang-hui,et al. Hierarchical data fusion publishing mechanism based on differential privacy protection[J]. Journal of Chinese Computer Systems,2019,40(10):2252 – 2256. (in Chinese)

[16] Chollet F. Xception:Deep learning with depth wise separable convolutions[A]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)[C]. USA: IEEE,2017. 1800 – 1807.

[17] Deep Learning Tutorials[OL]. http://deeplearning.net/tutorial/,2019-3-26.

作者简介

方 晨 男,1993 年出生,安徽安庆人. 战略支援部队信息工程大学博士研究生. 研究方向为机器学习隐私安全.
E-mail:17756230629@163.com

郭渊博(通信作者) 男,1975 年出生,陕西周至人. 战略支援部队信息工程大学教授、博士生导师. 研究方向为大数据安全、态势感知.
E-mail:yuanbo_g@hotmail.com

王 娜 女,1970 年出生,河南郑州人. 战略支援部队信息工程大学副教授. 研究方向为网络信息安全.