计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2019.20180353
56(9):1832-1842,2019

基于生成式对抗网络的结构化数据表生成模型

宋珂慧[1] 张 莹[1] 张江伟[2] 袁晓洁[1]
[1](南开大学计算机学院 天津 300350)
[2](新加坡国立大学计算机学院 新加坡 117417)
(songkehui@dbis.nankai.edu.cn)

A Generative Model for Synthesizing Structured Datasets Based on GAN

Song Kehui[1], Zhang Ying[1], Zhang Jiangwei[2], and Yuan Xiaojie[1]
[1](*College of Computer Science, Nankai University, Tianjin* 300350)
[2](*School of Computing, National University of Singapore, Singapore* 117417)

Abstract Synthesizing high quality dataset has been a long-standing challenge in both machine learning and database community. One of the applications of high quality dataset synthesis is to improve the model training, especially deep learning models. A robust model training process requires a large annotated dataset. One way of acquiring a large annotated training set is via the domain experts' manual annotation, which is expensive and prone to mistakes. Therefore, as an alternative, automatic synthesis of high quality and similar dataset is much more plausible. Some efforts have been devoted for synthesizing image dataset due to the rapid development of computer vision. However, those models can not be applied to the structured data (numeric & categorical table) directly. Moreover, little efforts have been payed to the numeric & categorical table. Therefore, we propose TableGAN, the first generative model from GAN family, which improves the performance of the generative model with adversarial learning mechanism. TableGAN modifies the internal structure of traditional GAN targeting numeric & categorical table, including the optimization function, to synthesize more high-quality training dataset samples for improving the effectiveness of the training models. Extensive experiments on real datasets show significant performance improvement for those models trained on the enlarged training datasets, and thus verify the effectiveness of our TableGAN.

Key words deep learning; generative models; neural network; generative adversarial network (GAN); classification

摘 要 在机器学习和数据库等领域,高质量数据集的合成一直以来是一个非常重要且充满挑战性的问题.其中,合成的高质量数据集可用来改善模型,尤其是深度学习模型的训练过程.一个健壮的模型训练过程需要大量已标注的数据集,获取这些数据集的一种方法是通过领域专家的手动标注,这种方法不仅代价大还容易出错,因此由模型自动合成高质量数据集的方法更为合理.近年来,由于计算机视觉领域的飞速发展,已经有不少致力于图像数据集合成的研究,但是这些模型不能直接应用在结构化数据表上,并且据调研,对这类数据的相关研究几乎没有.因此,提出了一个针对结构化数据表的生成模型TableGAN,该模型是生成式对抗网络(generative adversarial network, GAN)家族的一种变体,通过

收稿日期:2018-05-12;**修回日期**:2019-03-28
基金项目:国家自然科学基金项目(61772289,U1836109)
This work was supported by the National Natural Science Foundation of China (61772289, U1836109).
通信作者:张莹(yingzhang@nankai.edu.cn)

对抗训练的方式提高生成模型的性能.针对结构化数据的特征改变了传统 GAN 模型的内部结构,包括优化函数等,使其能够生成高质量的结构化数据用于改善模型的训练过程.通过在真实数据集上的大量实验表明了此模型的有效性,即在扩大后的数据集上训练模型的效果有明显提升.

关键词 深度学习;生成模型;神经网络;生成式对抗网络;分类

中图法分类号 TP301.6

近年来,在机器学习和数据库等领域,高质量数据集的合成问题一直以来是一个非常重要且充满挑战性的问题[1-2].合成的高质量数据集可用于很多场景,例如数据库性能基准测试(performance benchmarking)、降低数据挖掘成本以及改进模型训练过程等.其中,合成的高质量数据集可用来提升模型,尤其是深度学习模型的训练过程.

在训练某个机器学习模型的过程中,当训练样本数量不足时,很容易出现过拟合[3]现象.过拟合现象往往由训练样本数量不足引起,导致模型中的复杂参数只能捕捉训练样本中十分具体的随机特征,导致一些细微的误差都会对其产生巨大影响,因此在训练的过程中会出现模型在验证集上表现变差的现象.图 1 展示了分类器多层感知机(multi-layer perception, MLP)在数据集"Poker Hand"上的预测准确率曲线,从图 1 两条曲线的走向可以看出,在迭代 6 次之后,在训练集上的准确率尽管稳步上升,但在验证集上的准确率已经开始下降,也就是出现了过拟合现象,2 条曲线之间的区域大小反映了过拟合现象的严重程度.

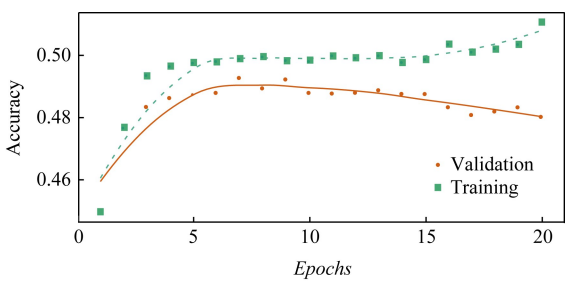


Fig. 1 An example of model performance

图 1 模型训练过程中的预测准确率

为了防止过拟合现象发生,需要将原有的训练集扩大.其中一种方法是领域专家手动标注更多的数据样本,但这既浪费人力又容易出错;另一种自动合成更多数据样本的方法更为可行.如图 2 所示,原始训练样本首先作为生成器(generator)的输入,生成器输出的合成训练样本和原始训练样本一起组成扩大后的训练集,最终将这个扩大后的训练集用于分类模型的训练.由于合成数据集质量较高且保留了原始数据样本中的重要特征,用扩大后的样本对分类模型进行训练的过程将更加稳定,并能够解决因训练样本不足引起的过拟合问题,提升了分类模型在验证集上的准确率.因此,设计一个性能良好的生成器是图 2 所示整个工作流程的重要环节.

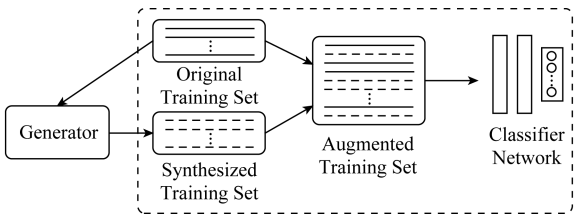


Fig. 2 The workflow of training classifiers using synthesized datasets

图 2 使用合成数据集训练分类模型的流程图

近年来,有不少与生成模型相关的研究[4-7],其中备受瞩目的是生成式对抗网络(generative adversarial network, GAN)[8].生成式对抗网络是 Goodfellow 等人[8]在 2014 年提出的一种生成模型,并被广泛应用于对原始样本分布特征的无监督式学习.目前为止,有不少针对 GAN 的相关研究,并衍生出若干 GAN 模型的变种,如 C-GAN[9]和 AC-GAN[10]等,都能够生成高质量的图片数据.

关系数据库中不具有主外键约束的单表被称为结构化数据表.结构化数据表包含若干属性,每个属性有自己特有的分布,属性间也有或强或弱的相关性,例如身高和体重正相关,身高越高的个体,体重就越大.属性的取值具有无序性(与结构化数据表中每条记录所处的位置无关)、取值离散等特点,与图片数据不尽相同.因此,GAN 及其若干变体都无法直接用于结构化数据表的生成.为了解决这个问题,本文主要提出了一个基于生成式对抗网络的结构化数据表生成模型,称为 TableGAN.

该模型为传统生成式对抗网络模型 GAN 的一种变体,由一个生成器(generator)模型 G 和一个判别器(discriminator)模型 D 组成.生成器 G 的目的是尽量学习原始数据的真实分布,生成让判别器甄别不出真伪的合成数据,而判别器 D 的目的是尽量

提升自己甄别原始数据与合成数据的判别能力.2个模型在相互对抗优化的过程中,不断提升各自的生成能力与判别能力.最终,生成器能够生成符合原始数据分布特征的合成数据,和原始数据一起用于分类模型的训练,从而解决由于训练样本不足导致的过拟合问题.和其他传统生成式对抗网络不同的是,TableGAN修改了优化函数,保证了模型有一个稳定的训练过程,并且为了防止噪声对模型稳定性的影响,在生成器模型和判别器模型中都添加了L_2正则化项,还增大了输入噪声的多样性,在一定程度上避免了模式崩溃(mode collapse)情况的发生.据我们所知,TableGAN模型是生成式对抗网络在结构化数据表生成领域的首次应用.

为了证明TabelGAN的有效性,本文提供了在2个数据集上,针对3种分类器网络的一系列实验结果和相关分析.充分的实验表明TableGAN能够生成有助于提升分类器网络训练的数据样本.为了更好地展示TableGAN生成数据的效果,我们选择了一个在数据挖掘比赛网站Kaggle①上排名最靠前的分类模型,实验证明使用合成的数据集训练后,分类模型的准确率仍可以进一步提升.

1 相关工作

数据合成在机器学习和数据库等领域有着十分重要的应用[11-13].其中一个在机器学习领域的应用就是利用合成的数据来解决过拟合问题.过拟合问题在机器学习领域存在已久,是一个亟待解决的问题.近年来,有不少学者提出对这个问题的解决方案,包括合成更多的训练样本[14]、交叉验证(cross-validation)[15]、正则化(regularization)[16]和提前停止(early stopping)[17]等方法.其中,合成更多的训练样本是最常使用的方法之一.

在计算机视觉领域,合成更多训练样本这一技术通常被称为数据增强(data augmentation).为了得到更多的训练样本,需要对原始训练图像进行简单的几何和外观方面的转换,包括对图片进行旋转、扭曲等,但是这些转换都基于一个很强的假设,即这些细微的物理转换都不会改变图片的类别标签.由于此假设没有相关的理论证明,这种通过物理转换来扩大训练集的方法具有一定的局限性.

生成模型是近年来机器学习领域最有前景的方法之一,它通过学习并遵从给定数据集的概率分布来生成新的样本数据.其中变分自动编码器(variational auto-encoders, VAE)[6]和生成式对抗网络(GAN)[8]是生成模型中众所周知的代表.

VAE是一个概率图模型,由一个编码器(encoder)和一个解码器(decoder)构成,编码器将数据分布的高级特征映射到数据的低级表征(latent vector),解码器接受数据的低级表征,然后输出同样数据的高级表征.VAE的训练过程完全依赖于一个假设损失函数及KL散度,使得生成的数据尽可能去接近真实数据的分布.

然而,GAN为我们提供了一个对目标函数更为灵活的定义,其中包括Jensen-Shannon[8]、所有的f-divergences[18]以及一些其他距离度量的组合[19].GAN由一个生成器G和一个判别器D组成,它们均由深度学习网络实现.生成器和判别器相互对抗进行训练,生成器尽可能生成与原始数据分布相近的数据集,使判别器无法将其与原始数据区分,而判别器则尽可能提升自己区分原始数据与合成数据的能力.经过一段时间的对抗训练后,生成器能够生成接近原始数据分布的样本,用于解决由于训练样本不足导致的过拟合问题.GAN被证明训练难度大且十分不稳定[20],因此不少学者提出了GAN的若干变体,用于改进生成数据的质量.例如,C-GAN[9]将条件信息,即类标签,添加到生成器模型输入中,用于改进原始GAN模型.AC-GAN[10]中的判别器不仅要判别输入数据来自原始数据还是合成数据,还要判别输入数据的类别标签.本文提出了GAN模型的另一个变体TableGAN,用于生成高质量的结构化数据表,并将其用来训练分类模型以改善模型的训练过程.

2 算法实现

本节主要介绍文中所提出算法的模型推导和理论分析,首先对模型训练过程发生的过拟合现象进行形式化定义和描述,然后回顾生成式对抗网络的基本原理,最后给出基于GAN的结构化数据表生成模型TableGAN中算法的相关理论分析,包括模型推导、算法伪代码等.

2.1 问题定义

给定一个带标签的训练集$Y=\{\boldsymbol{y}_n\}^N$,其中

① https://www.kaggle.com/c/sf-crime/discussion/15836

$\boldsymbol{y}_n=(\boldsymbol{x}_n,c_n)$,$c_n\in\{1,2,\cdots,M\}$是第 n 行数据的标签,$\boldsymbol{x}_n$ 是除了标签之外的其他属性.训练一个神经网络的基本目标是,用给定训练集去估计模型中的所有参数:

$$\boldsymbol{\theta}^*=\arg\max_{\boldsymbol{\theta}}\log p(\boldsymbol{\theta}|\boldsymbol{y}),\tag{1}$$

结合贝叶斯公式:

$$p(\boldsymbol{\theta}|\boldsymbol{y})=p(\boldsymbol{\theta}|\boldsymbol{x},c)\propto p(\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta})p(c|\boldsymbol{x},\boldsymbol{\theta}).\tag{2}$$

假设所有的训练样本均为条件独立,可以得到:

$$\log p(\boldsymbol{\theta}|\boldsymbol{y})\approx\log p(\boldsymbol{\theta})+\frac{1}{N}\sum_{n=1}^{N}(\log p(\boldsymbol{x}_n|\boldsymbol{\theta})+\log p(c_n|\boldsymbol{x}_n,\boldsymbol{\theta})),\tag{3}$$

其中,$p(\boldsymbol{\theta})$为模型所有参数的先验概率,$p(\boldsymbol{x}_n|\boldsymbol{\theta})$是对样本 $\boldsymbol{x}_n$ 的似然估计,$p(c_n|\boldsymbol{x}_n,\boldsymbol{\theta})$是对标签 c_n 在给定 $\boldsymbol{x}_n$ 和 $\boldsymbol{\theta}$ 条件下的似然估计.

在训练神经网络时,模型中所有参数通过梯度下降的方式找到最优解.然而,当训练样本 Y 数量不足时,往往会出现过拟合现象.也就是说,尽管模型在训练集上效果很好,但在验证集上效果却很差.因此,我们需要合成更多高质量的训练样本,这些新合成的样本需要保留原始训练样本的重要特征,使扩大后的样本能够更好地训练模型中的参数.本文提出了一个基于生成式对抗网络的结构化数据表生成模型——TableGAN,用来扩大原有的训练样本并保留原始样本中的重要特征,为后续神经网络的稳定训练提供良好保障.

2.2 生成式对抗网络 GAN

生成式对抗网络 GAN 是 Goodfellow 等人[8]在 2014 年提出的一种生成模型,目前已经成为人工智能学界一个热门的研究方向.GAN 的基本思想源于博弈论中的二人零和博弈,即二人的利益之和为零,一方所得正好为另一方所失.因此,GAN 由 2 个相互博弈的神经网络模型组成,一个叫生成器 G,另一个叫判别器 D.生成器 G 的目的是尽量学习原始数据的真实分布,生成让判别器甄别不出真伪的合成数据;而判别器 D 的目的是尽量提升自己甄别原始数据与合成数据的判别能力.2 个模型在相互对抗优化的过程中,不断提升各自的生成能力与判别能力,这个学习优化过程就是寻找二者之间的一个纳什均衡.在训练优化一段时间之后,生成式对抗网络的生成器能够捕捉原始数据的真实分布,并生成一系列符合同一分布的合成数据样本.

生成器为了捕捉原始数据 $\boldsymbol{x}$ 的真实分布 p_g,使用一个映射函数(一般由深度神经网络实现),将一个已知的分布 $p(\boldsymbol{z})$,例如高斯分布,映射到另一个数据空间 $G(\boldsymbol{z},\boldsymbol{\theta}_g)$,其中 $\boldsymbol{z}$ 称之为噪声(noise),$\boldsymbol{\theta}_g$ 表示生成器模型中的所有参数.生成器的目标是尽量缩小 $G(\boldsymbol{z},\boldsymbol{\theta}_g)$与真实数据分布 $p_{\text{data}}(\boldsymbol{x})$之间的差异.对于判别器模型来说,通过输出 0 或 1 来表示判别器对输入数据真假的判别情况.当输入数据采样于原始数据 $p_{\text{data}}(\boldsymbol{x})$时,判别器输出为 1;而当输入数据采样于合成数据集 $G(\boldsymbol{z})$,也就是从生成器中输出的数据时,判别器输出为 0.

在 GAN 的训练过程中,生成器模型和判别器模型进行相互对抗来进行优化,因此对 G 和 D 进行交替式训练.对于 G 而言,需要最小化 $\log(1-D(G(\boldsymbol{z})))$,也就是尽可能让 G 合成的数据集 $G(\boldsymbol{z})$能够欺骗 D,使得判别器 D 的输出 $D(G(\boldsymbol{z}))$接近 1.然而对判别器 D 而言,需要增强自己判别真假数据的能力,即最大化 $\log D(\boldsymbol{x})$与 $\log(1-D(G(\boldsymbol{z})))$,也就是当输入数据为真实数据 $\boldsymbol{x}$ 时,判别器的输出 $D(\boldsymbol{x})$尽可能接近 1,而当输入数据为合成数据 $G(\boldsymbol{z})$时,判别器的输出 $D(G(\boldsymbol{z}))$尽可能接近 0.因此,GAN 的优化问题是一个极小-极大化问题,GAN 的目标函数可以描述为

$$\min_G\max_D V(D,G)=E_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]+E_{\boldsymbol{z}\sim p(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z})))].\tag{4}$$

2.3 结构化数据表生成模型 TableGAN

本节主要介绍基于 GAN 的结构化数据表生成模型 TableGAN.图 3 给出了模型 TableGAN 的示意图,TableGAN 由一个生成器 G 和一个判别器 D 组成,符合某种分布的噪声 $\boldsymbol{z}$ 与类标签 $\boldsymbol{c}$ 一起作为生成器 G 的输入,经过 G 的变换后生成合成数据样本 $G(\boldsymbol{z}|\boldsymbol{c})$,随后与真实数据样本 $\boldsymbol{x}$ 一起作为判别器

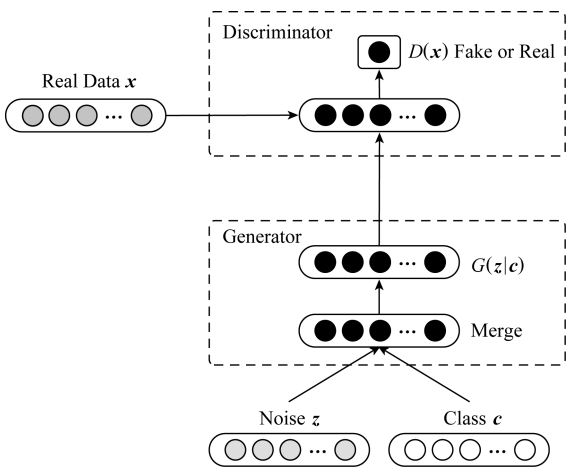


Fig. 3 The structure of our TableGAN

图 3 TableGAN 模型示意图

D 的输入,判别器的最终输出又会进一步指导生成器网络的训练过程.

生成器网络与判别器网络均由深度神经网络实现,生成器网络和判别器网络中的所有参数分别由 $\boldsymbol{\theta}$ 与 $\boldsymbol{\gamma}$ 表示.2 个网络相互对抗进行训练,目标函数为

$$\min_G \max_D V(D,G)=E_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]+ E_{\boldsymbol{z}\sim p(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z}|\boldsymbol{c})))]. \tag{5}$$

式(5)与传统 GAN 模型的目标函数对比而言,增加了类别标签 $\boldsymbol{c}$ 作为生成器的输入,即给生成器额外的信息指导其更好地生成数据.然而在训练的过程中,使用式(5)作为目标函数易出现生成器梯度消失现象,从而导致模型极难训练,文献[21]中有相关理论证明.因此,TableGAN 模型使用 Earth-Mover(EM)距离来衡量原始样本与合成样本之间的距离,即使 2 个分布没有重叠或重叠的部分非常少,依然能够反映 2 个分布的远近,EM 距离定义为

$$W(P_1,P_2)=\inf_{\gamma\sim\Pi(P_1,P_2)} E_{(\boldsymbol{x},\boldsymbol{y})\sim\gamma}[\|\boldsymbol{x}-\boldsymbol{y}\|], \tag{6}$$

其中 $\Pi(P_1,P_2)$ 为 P_1 和 P_2 所有可能的联合分布,计算在此联合分布下样本对距离的期望,此期望的下界就是 EM 距离.因此,使用 EM 距离后的目标函数为

$$\min_G \max_D V(D,G)=E_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[D_{\boldsymbol{\gamma}}(\boldsymbol{x})]- E_{\boldsymbol{z}\sim p(\boldsymbol{z})}[D_{\boldsymbol{\gamma}}(G_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c}))]. \tag{7}$$

传统 GAN 模型在训练过程中往往会发生模式崩溃(mode collapse)的现象,这指的是模型只能捕捉并保留原始数据中很少的一部分特征,以致生成的数据样本十分单一.我们的 TableGAN 则针对这个问题,使用 3 个技巧来缓解模式崩溃的现象:1)增加生成器输入噪声 $\boldsymbol{z}$ 的多样性.对图片数据集来说,传统 GAN 模型生成器的输入噪声服从单峰的正态分布,而对于本文需要生成的结构化数据表来说,输入多峰分布的噪声能够增加合成数据的多样性;2)我们放弃基于动量的优化方法,例如 Adam,而使用 RMSProp[22-23];3)在神经网络模型上增加 L_2 正则化项,保证 TableGAN 训练过程中的稳定.

TableGAN 的训练过程如算法 1 所示,针对参数 $\boldsymbol{\theta}$ 与 $\boldsymbol{\gamma}$,使用式(7)给出的目标函数来分别交替训练生成器网络与判别器网络,训练过程收敛后会得到:

$$\boldsymbol{\gamma}^*=-\arg\min_{\boldsymbol{\gamma}} V(D,G). \tag{8}$$

此时,判别器 $D_{\boldsymbol{\gamma}^*}$ 已经收敛,$\boldsymbol{\theta}^*$ 也已收敛于 $V(D,G)$ 的最小值,模型已经训练至稳定状态.之后,我们使用模型中已训练好的生成器,生成更多的训练样本,用于分类模型的训练过程.

算法 1. TableGAN 训练算法.

输入:学习率(learning rate)η、剪切参数(clipping parameter)d、批大小(batch size)m、生成器每迭代 1 次时判别器迭代的次数 n_d;

输出:收敛后生成器网络和判别器网络的参数 $\boldsymbol{\theta}$ 与 $\boldsymbol{\gamma}$.

① WHILE 不收敛 DO
② FOR $t=0,1,2,\cdots,n_d$
③ 采样 $\{\boldsymbol{z}^{(i)}\}_{i=1}^m\sim p(\boldsymbol{z})$;
④ 采样 $\{\boldsymbol{x}^{(i)}\}_{i=1}^m\sim p_{\text{data}}(\boldsymbol{x})$;
⑤ $g_{\boldsymbol{\gamma}}\leftarrow-\nabla_{\boldsymbol{\gamma}}\left[\frac{1}{m}\sum_{i=1}^m D_{\boldsymbol{\gamma}}(\boldsymbol{x}^{(i)})-\frac{1}{m}\sum_{i=1}^m D_{\boldsymbol{\gamma}}(G_{\boldsymbol{\theta}}(\boldsymbol{z}^{(i)}|\boldsymbol{c}))\right]$;
⑥ $\boldsymbol{\gamma}\leftarrow\boldsymbol{\gamma}-\eta\times RMSProp(\boldsymbol{\gamma},g_{\boldsymbol{\gamma}})$;
⑦ $\boldsymbol{\gamma}\leftarrow clip(\boldsymbol{\gamma},-d,d)$;
⑧ END FOR
⑨ 采样 $\{\boldsymbol{z}^{(i)}\}_{i=1}^m\sim p(\boldsymbol{z})$;
⑩ $g_{\boldsymbol{\theta}}\leftarrow-\nabla_{\boldsymbol{\theta}}\frac{1}{m}\sum_{i=1}^m D_{\boldsymbol{\gamma}}(G_{\boldsymbol{\theta}}(\boldsymbol{z}^{(i)}|\boldsymbol{c}))$;
⑪ $\boldsymbol{\theta}\leftarrow\boldsymbol{\theta}-\eta\times RMSProp(\boldsymbol{\theta},g_{\boldsymbol{\theta}})$;
⑫ END WHILE

3 实验与分析

本节主要介绍相关实验设置,包括实验所使用的数据集、分类模型以及用于比较的基准算法,之后给出实验结果并对其进行分析与讨论.实验代码已更新至 GitHub①.

针对每个数据集,我们采取 3 个实验步骤:

1) 使用原始训练样本对分类模型进行训练,在测试集上得到分类模型预测准确率;

2) 使用原始训练样本,对数据库领域结构化数据表扩展方法Dscaler、数据匿名化方法k-anonymity与 t-closeness、生成式对抗网络 C-GAN 和我们的模型 TableGAN 进行训练,随后使用训练好的模型生成合成的数据集,与原始训练样本一起组成了扩大后的数据集;

① https://github.com/cocoissong/TableGAN

3) 使用步骤 2 中扩大后的数据集进行训练,在测试集上得到分类模型的预测准确率,和步骤 1 中得到的准确率进行比较.

3.1 数据集

本文使用 2 个公开的数据集用于实验.一个是数据挖掘比赛网站 Kaggle 上公开的数据集①,另一个是机器学习仓库 UCI② 上公开的数据集,表 1 提供了 2 个数据集的统计信息.

Table 1 Summaries of the 2 Datasets

表 1 实验数据集统计信息

Dataset	Source	# Classes	# Samples
SF Crime	Kaggle	39	877 982
Poker Hand	UCI	10	1 025 010

1) SF Crime.本数据集收集了旧金山市近 12 年来的犯罪记录,共有 9 个不同的属性,其中属性"*Category*"为标签,共有 39 种不同的取值.分类模型需要根据犯罪事件发生的时间与地点来预测犯罪的种类.表 2 提供了此数据集的详细信息.

Table 2 Summaries of the SF Crime Dataset

表 2 关于 SF Crime 数据集的描述

Attribute	Description	Data Type
Dates	Timestamp of the Crime Incident	Datetime
Category	Category of the Crime Incident	Categorical
Description	Detailed Description of the Crime Incident	Text
DayOfWeek	Day of the Week	Number
PdDistrict	Name of the Police Department District	Categorical
Resolution	How the Crime Incident was Resolved	Categorical
Address	Approximate Street Address of the Crime Incident	Categorical
X	Longitude	Number
Y	Latitude	Number

2) Poker Hand.本数据集记录了从 52 张扑克牌中抽出 5 张扑克牌的大小与花色,共有 11 个不同的属性,其中属性"*Class*"为标签,共有 10 种不同的取值,包括"同花顺"、"同花"、"顺子"等.分类模型需要根据 5 张扑克牌的大小与花色来预测牌型.表 3 提供了此数据集的详细信息.

Table 3 Summaries of the Poker Hand Dataset

表 3 关于 Poker Hand 数据集的描述

Attribute	Description
S_1	Suit of card # 1, ordinal (1-4) representing {Hearts, Spades, Diamonds, Clubs}
C_1	Rank of card # 1, numerical (1-13) representing (Ace, 2, 3, …, Queen, King)
⋮	⋮
S_5	Suit of card # 5
C_5	Rank of card # 5
Class	Class "Poker Hand", ordinal (0-9)

3.2 分类模型

本文使用 3 个公开的分类模型,包括 1 个性能良好的多层感知机(MLP),以及 2 个经典的分类算法——随机森林(random forest, RF)和决策树(decision tree, DT).

1) MLP.这是引言提到的在数据挖掘比赛网站 Kaggle③ 上排名最靠前的分类模型,它是一个 3 层神经元感知器,在 SF Crime 数据集下,这个分类模型的性能在所有的公开算法中排名前 1%.

2) RF.随机森林是通过集成学习的思想将多棵树集成的一种算法,它的基本单元是决策树,而它的本质属于机器学习的一大分支——集成学习(ensemble learning)方法,其输出的类别由个别树输出的类别的众数而定.也就是说,对于一个输入样本,N 棵树会有 N 个分类结果,而随机森林集成了所有的分类投票结果,将投票次数最多的类别指定为最终的输出.

3) DT.决策树是一种基本的分类方法.决策树模型呈树形结构,表示基于特征对实例进行分类的过程.它可以认为是 if-then 规则的集合,也可以认为是定义在特征空间与类空间上的条件概率分布,具有可读性、效率高等优点.

本文模型 TableGAN 由高层神经网络 API——Keras 来实现,基于 TensorFlow 后端.针对每个数据集,TableGAN 根据 *Epochs* 和 *D_iters* 这 2 个参数的不同取值,生成 17 份不同的合成数据样本.其中,*Epochs* 反映了模型的学习程度,如果训练时的 *Epochs* 过小,由于特征学习不够充分,生成的合成数据集不足以大幅提高分类模型的预测准确率,

① https://www.kaggle.com/c/sf-crime/data

② http://archive.ics.uci.edu/ml/datasets/Poker+Hand

③ https://www.kaggle.com/c/sf-crime/discussion/15836

反之,如果 $Epochs$ 过大,模型会学习数据中过于具体的特征,依旧会影响分类模型的预测准确率,本实验 $Epochs$ 的取值在 20～90 之间.D_iters 反映了模型中判别器相对于生成器的迭代次数,即每当生成器迭代 1 次时判别器迭代的次数.例如 $D\_iters=5$ 表明每当模型生成器训练 1 次时判别器训练 5 次.此参数表明维持生成器和判别器这 2 个模型训练程度的动态平衡具有十分重要的意义.

3.3 基准算法

本文采用 10 折交叉验证的方式对提出的 TableGAN 算法和 4 个方法在 2 个数据集上进行了实验,并将结果进行了比较和分析.

1) Without scaling up. 未采用任何生成模型,使用原始训练样本对分类模型进行训练.

2) Dscaler[24]. 数据库领域较新的结构化数据扩展方法 Dscaler,一般针对多张具有主外键关系的结构化数据表,旨在保留主外键间参照关系.而单个结构化数据表的扩展方法,只是简单在数据表中进行采样,以此合成新的数据集.

3) Anonymization. 采用数据匿名化方法 k-anonymity 与 t-closeness 结合.参数 $k\in\{2,10,100\}$,$t\in\{0.001,0.1,0.5\}$,表 4 的实验结果取这些参数下最高的准确率值.

4) C-GAN[9]. C-GAN 是传统生成式对抗网络的一种变体,通过增加额外信息来提升合成数据的质量.其在图片数据集 MNIST 上表现良好,能够根据标签生成高质量的图片.

5) TableGAN 为本文提出的算法.

3.4 实验结果分析

本节通过比较使用扩大后的训练集与原始训练集对分类模型的训练情况来证明 TableGAN 的有效性.我们使用训练后的分类模型在验证集上的预测准确率来量化 TableGAN 合成数据的质量.表 4 呈现了在 2 个数据集上的所有实验结果.可以看出,TableGAN 在大部分情况下都可以改进分类模型的训练情况,并且比 Dscaler,Anonymization,C-GAN 这 3 个模型表现要好.3.4.1 和 3.4.2 节有对实验结果详细的对比分析,并根据 $Epochs$ 和 D_iters 这 2 个参数的变化情况绘制了分类模型对应的预测结果图.

Table 4 Quantitative Results on the 17 Versions of Training Data

表 4 分类模型在 17 个不同版本的训练集下的所有结果列表

Datasets	Classifiers	$Epochs$ ($D\_iters=6$)								D_iters ($Epochs=30$)									
		20	30	40	50	60	70	80	90	1	2	3	4	5	6	7	8	9	10
SF Crime	MLP	27.84	27.84	27.84	27.84	27.84	27.84	27.84	27.84	27.84	27.84	27.84	27.84	27.84	27.84	27.84	27.84	27.84	27.84
	MLP♦	25.95	25.95	25.95	25.95	25.95	25.95	25.95	25.95	25.95	25.95	25.95	25.95	25.95	25.95	25.95	25.95	25.95	25.95
	MLP✍	27.86	27.86	27.86	27.86	27.86	27.86	27.86	27.86	27.86	27.86	27.86	27.86	27.86	27.86	27.86	27.86	27.86	**27.86**
	MLP*	27.83	27.65	27.56	27.24	27.52	27.67	27.67	27.97	27.89	27.62	**28.02**	27.91	27.84	27.65	27.64	27.91	**27.89**	27.24
	MLP▽	**27.97**	**28.05**	**27.88**	**28.41**	**28.08**	**28.18**	**28.12**	**28.28**	**27.97**	**28.06**	28.01	**28.03**	**28.11**	**28.05**	**28.27**	**28.22**	27.81	27.56
	RF	23.33	23.33	23.33	23.33	23.33	23.33	23.33	23.33	23.33	23.33	23.33	23.33	23.33	23.33	23.33	23.33	23.33	23.33
	RF♦	23.11	23.11	23.11	23.11	23.11	23.11	23.11	23.11	23.11	23.11	23.11	23.11	23.11	23.11	23.11	23.11	23.11	23.11
	RF✍	23.35	23.35	23.35	23.35	23.35	23.35	23.35	23.35	23.35	23.35	23.35	23.35	23.35	23.35	23.35	23.35	**23.35**	**23.35**
	RF*	23.36	**23.41**	23.40	**23.44**	23.43	23.40	23.39	23.40	23.28	23.34	23.31	23.31	23.30	23.31	**23.35**	23.33	23.30	23.27
	RF▽	**23.37**	23.39	**23.44**	23.41	**23.49**	**23.44**	**23.46**	**23.47**	**23.48**	**23.45**	**23.35**	**23.52**	**23.49**	**23.39**	23.26	**23.50**	23.34	23.31
	DT	18.3	18.3	18.3	18.3	18.3	18.3	18.3	18.3	18.3	18.3	18.3	18.3	18.3	18.3	18.3	18.3	18.3	18.3
	DT♦	18.33	18.33	18.33	18.33	18.33	18.33	18.33	18.33	18.33	18.33	18.33	18.33	18.33	18.33	18.33	18.33	18.33	18.33
	DT✍	18.37	18.37	18.37	**18.37**	18.37	**18.37**	18.37	18.37	18.37	18.37	18.37	**18.37**	18.37	18.37	18.37	18.37	**18.37**	18.37
	DT*	18.33	18.36	**18.38**	18.32	18.30	18.33	18.37	18.37	18.33	18.35	18.31	18.36	18.38	18.36	18.29	18.28	18.30	18.31
	DT▽	**18.41**	**18.39**	18.37	18.32	**18.38**	18.36	**18.40**	**18.40**	**18.38**	**18.41**	**18.40**	18.34	**18.39**	**18.39**	**18.39**	**18.47**	18.36	**18.49**
Poker Hand	MLP	54.71	54.71	54.71	54.71	54.71	54.71	54.71	54.71	54.71	54.71	54.71	54.71	54.71	54.71	54.71	54.71	54.71	54.71
	MLP♦	51.71	51.71	51.71	51.71	51.71	51.71	51.71	51.71	51.71	51.71	51.71	51.71	51.71	51.71	51.71	51.71	51.71	51.71
	MLP✍	**55.82**	55.82	55.82	55.82	55.82	55.82	55.82	**55.82**	**55.82**	**55.82**	55.82	55.82	55.82	55.82	55.82	55.82	55.82	55.82
	MLP*	51.55	52.45	51.51	52.66	53.22	51.84	52.15	52.65	53.63	53.52	50.84	52.77	52.58	52.45	51.66	54.16	51.28	51.27
	MLP▽	53.64	**60.16**	**56.05**	**56.27**	**56.87**	**57.45**	**56.34**	55.71	55.57	55.56	**56.94**	**56.77**	**56.51**	**60.16**	**56.63**	**56.76**	**56.81**	**56.55**

Continued (Table 4)

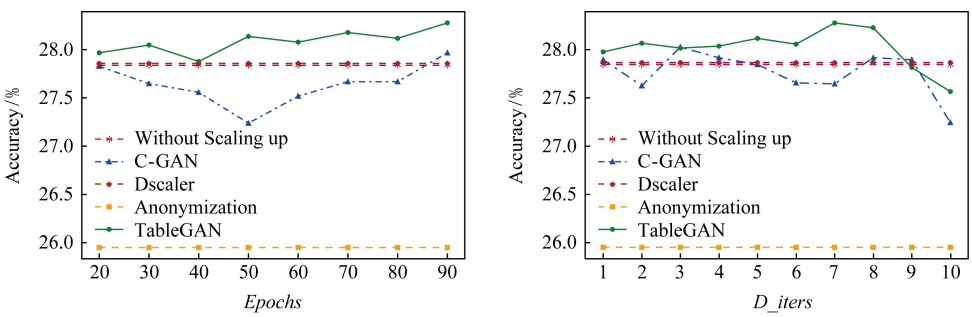
Datasets	Classifiers	*Epochs* (*D_iters*=6)								*D_iters* (*Epochs*=30)									
		20	30	40	50	60	70	80	90	1	2	3	4	5	6	7	8	9	10
Poker Hand	RF	56.08	56.08	56.08	56.08	56.08	56.08	56.08	56.08	56.08	56.08	56.08	56.08	56.08	56.08	56.08	56.08	56.08	56.08
	RF♦	56.67	56.67	56.67	56.67	56.67	56.67	56.67	56.67	56.67	56.67	56.67	56.67	56.67	56.67	56.67	56.67	56.67	**56.67**
	RF✍	55.77	55.77	55.77	55.77	55.77	55.77	55.77	55.77	55.77	55.77	55.77	55.77	55.77	55.77	55.77	55.77	55.77	55.77
	RF*	55.08	56.62	56.39	56.48	56.04	56.10	56.12	55.48	55.11	54.57	56.31	54.98	55.47	56.62	55.67	55.85	55.30	55.38
	RF☞	**56.36**	**57.11**	**56.66**	**57.56**	**57.24**	**57.24**	**56.29**	**57.02**	**56.82**	**57.68**	**57.20**	**57.11**	**57.07**	**57.11**	**57.21**	**57.59**	**57.31**	56.57
	DT	47.86	47.86	47.86	47.86	47.86	47.86	47.86	47.86	47.86	47.86	47.86	47.86	47.86	47.86	47.86	47.86	47.86	47.86
	DT♦	47.98	47.98	47.98	47.98	47.98	47.98	47.98	47.98	47.98	47.98	47.98	47.98	47.98	47.98	47.98	47.98	47.98	47.98
	DT✍	47.91	47.91	47.91	47.91	47.91	47.91	47.91	47.91	47.91	47.91	47.91	47.91	47.91	47.91	47.91	47.91	47.91	47.91
	DT*	47.99	47.87	51.66	50.85	50.22	50.97	51.48	50.70	47.72	50.03	51.42	47.79	47.87	47.87	50.85	47.90	47.95	47.80
	DT☞	**52.73**	**52.54**	**52.70**	**52.68**	**52.45**	**52.57**	**52.61**	**52.65**	**52.55**	**52.63**	**52.52**	**52.57**	**52.62**	**52.54**	**52.64**	**52.58**	**52.51**	**52.58**

Notes: "♦" means the corresponding classifiers using the augmented training data produced by data anonymization algorithms (k-anonymity + t-closeness); "✍" means the classification results of data produced by Dscaler; "*" means the classification results of data produced by C-GAN; "☞" means the classification results of data produced by our TableGAN. The best results have been highlighted in bold.

3.4.1 SF Crime 数据集上效果对比

图 4 展示了在 SF Crime 数据集上应用分类模型 MLP 的实验结果.其中,TableGAN 的性能一直优于 C-GAN 的性能,即使这个分类模型已经是在此数据集下性能排名前 1%的分类器,TableGAN 依旧可以通过扩大训练样本的方式,进一步提升分类模型的预测准确率.而数据隐私算法扩大后的数据集,由于隐藏数据中部分重要特征,训练分类模型的准确率还不如原始训练样本对分类模型进行训练的准确率.

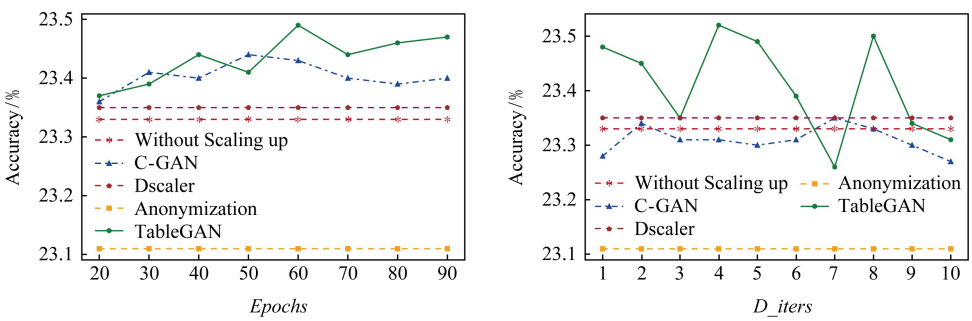
图 5 和图 6 分别展示了在分类模型随机森林和决策树下的实验结果.尽管这 2 个传统分类模型的



(a) Accuracy when *D_iters*=6 (b) Accuracy when *Epochs*=30

Fig. 4 Performance comparison using MLP classifier on SF Crime dataset

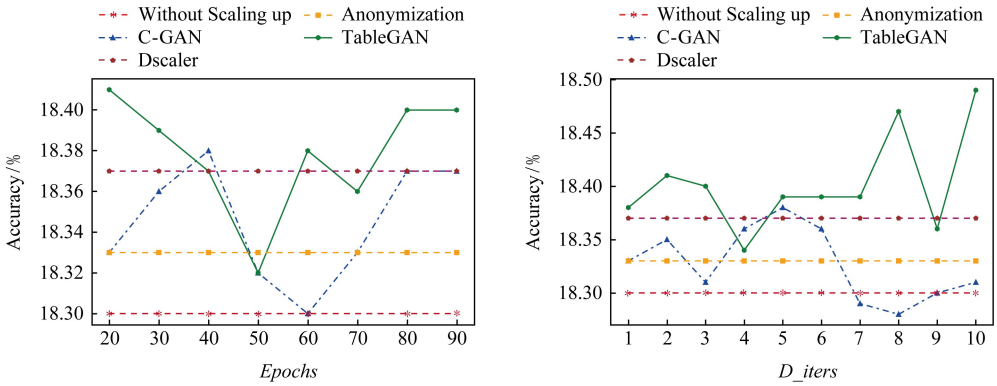
图 4 使用 MLP 在数据集 SF Crime 上的性能对比



(a) Accuracy when *D_iters*=6 (b) Accuracy when *Epochs*=30

Fig. 5 Performance comparison using Random Forest classifier on SF Crime dataset

图 5 使用随机森林在数据集 SF Crime 上的性能对比



(a) Accuracy when *D_iters*=6 (b) Accuracy when *Epochs*=30

Fig. 6 Performance comparison using Decision Tree classifier on SF Crime dataset

图 6 使用决策树在数据集 SF Crime 上的性能对比

学习能力不如 MLP 强,也就是过拟合现象不够显著,但 TableGAN 依旧能够提升分类模型的准确率,TableGAN 的表现也优于 C-GAN 模型的表现.

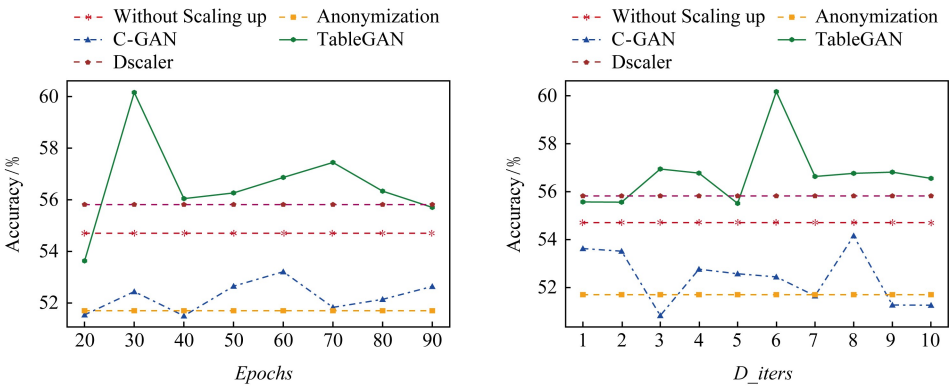
为更好地证明本文方法 TableGAN 在数据集 SF Crime 上的优越性,使用配对样本 t 检验.显著性检验表明,TableGAN 在置信区间为 0.95 的情况下,性能优于其他所有算法.

3.4.2 Poker Hand 数据集上效果对比

图 7 展示了在 Poker Hand 数据集上应用分类模型 MLP 的实验结果.可以看出使用 TableGAN 扩大原始训练样本之后能够大幅提升分类模型的准确率,并且 TableGAN 比 C-GAN 有着更好的性能.当 TableGAN 训练 30 轮,且每当生成器训练一次后判别器被训练 6 次时,TableGAN 提升分类模型的性能最显著,准确率由原来的 54.71%提升至 60.16%.通过观察分类模型训练过程中的 loss 曲线,使用 TableGAN 扩大训练样本在很大程度上缓解了过拟合的问题.

图 8 和图 9 分别展示了在分类模型随机森林和决策树下的实验结果.使用 TableGAN 扩大训练样本后,能将分类模型随机森林的准确率由原来的 56.08%提升至 57.68%,并能将分类模型决策树的准确率由原来的 47.86%提升至 52.73%.从图 8 和图 9 可以看出 TableGAN 很大程度上提升了分类模型的预测准确率,并总比使用 C-GAN 的性能好.从图 9 可以看出,随着参数 *Epochs* 和 *D_iters* 的变化,分类模型的预测准确率变化不大(最上方的曲线较为平缓),也就是说,我们的模型 TabelGAN 即使没有谨慎选择参数,仍然可以生成高质量的合成数据集来改善分类模型的训练过程,反观 C-GAN,参数的细微变化很大程度上影响了分类模型的准确率.

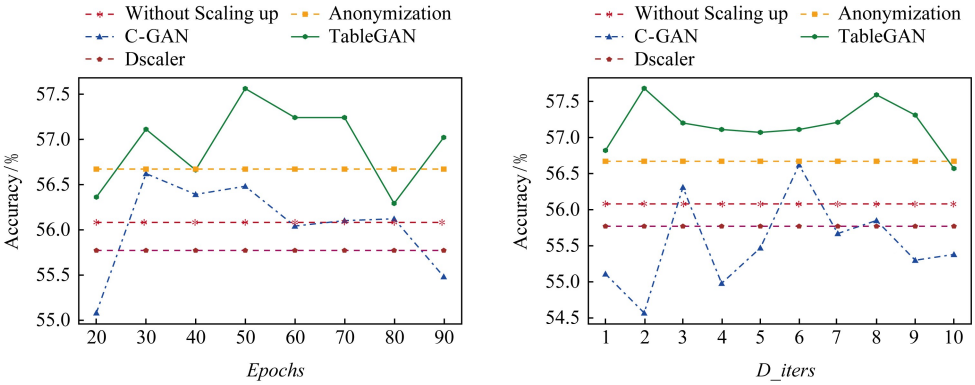
为更好地证明本文模型 TableGAN 在数据集 Poker Hand 上的优越性,使用配对样本 t 检验.显著性检验表明,TableGAN 在置信区间为 0.95 的情况下,性能优于其他所有算法.



(a) Accuracy when *D_iters*=6 (b) Accuracy when *Epochs*=30

Fig. 7 Performance comparison using MLP classifier on Poker Hand dataset

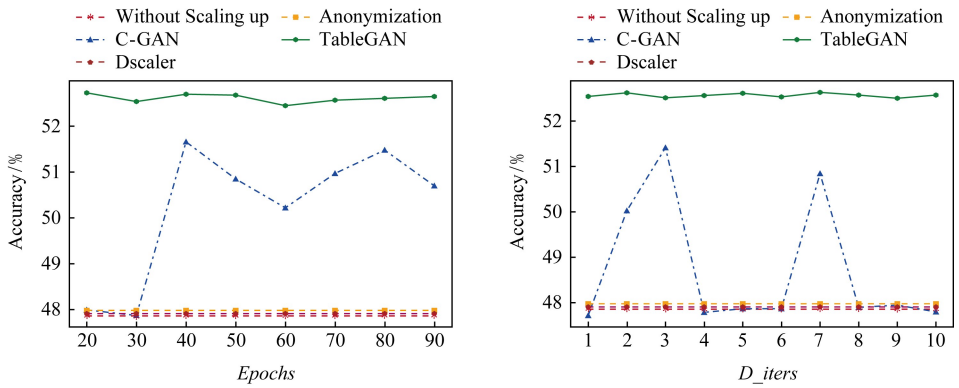
图 7 使用 MLP 在数据集 Poker Hand 上的性能对比



(a) Accuracy when *D_iters*=6 (b) Accuracy when *Epochs*=30

Fig. 8 Performance comparison using Random Forest classifier on Poker Hand dataset

图 8 使用随机森林在数据集 Poker Hand 上的性能对比



(a) Accuracy when *D_iters*=6 (b) Accuracy when *Epochs*=30

Fig. 9 Performance comparison using Decision Tree classifier on Poker Hand dataset

图 9 使用决策树在数据集 Poker Hand 上的性能对比

总之,通过实验可以看出我们的模型 TableGAN 在 2 个数据集上都能够生成高质量的合成数据,用于改善分类模型的训练过程,从而提升分类模型的预测准确率.

4 总结与工作展望

本文研究了结构化数据表的生成问题,提出一个基于生成式对抗网络的生成模型,生成符合原始数据样本分布的合成样本,以扩大训练样本的方式解决由于训练样本不足导致的分类模型过拟合问题.实验证明,本文提出的方法能够生成高质量的结构化数据表,进一步提高分类模型的准确率.

参 考 文 献

[1] Tay Y C, Dai B T, Wang D T, et al. UpSizeR: Synthetically scaling an empirical relational database [J]. Information Systems, 2013, 38(8): 1168-1183

[2] Odena A, Olah C, Shlens J. Conditional image synthesis with auxiliary classifier gans [J]. arXiv preprint arXiv:1610.09585, 2016

[3] Hawkins D M. The problem of overfitting [J]. Journal of Chemical Information and Computer Sciences, 2004, 44(1): 1-12

[4] Vincent P, Larochelle H, Bengio Y, et al. Extracting and composing robust features with denoising autoencoders [C] // Proc of the 25th Int Conf on Machine Learning. New York: ACM, 2008: 1096-1103

[5] Eslami S M A, Heess N, Williams C K I, et al. The shape Boltzmann machine: A strong model of object shape [J]. International Journal of Computer Vision, 2014, 107(2): 155-176

[6] Kingma D P, Welling M. Auto-encoding variational Bayes [J]. arXiv preprint arXiv:1312.6114, 2013

[7] Mnih V, Susskind J M, Hinton G E. Modeling natural images using gated MRFs [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(9): 2206-2222

[8] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets [C] //Proc of Advances in Neural Information Processing Systems (NIPS). Cambridge, MA: MIT Press, 2014: 2672-2680

[9] Mirza M, Osindero S. Conditional generative adversarial nets [J]. arXiv preprint arXiv:1411.1784, 2014

[10] Odena A, Olah C, Shlens J. Conditional image synthesis with auxiliary classifier gans [J]. arXiv preprint arXiv:1610.09585, 2016

[11] Patki N, Wedge R, Veeramachaneni K. The synthetic data vault [C] //Proc of the 3rd IEEE Int Conf on Data Science and Advanced Analytics (DSAA). Piscataway, NJ: IEEE, 2016: 399-410

[12] Tay Y. Data generation for application-specific benchmarking [C] //Proc of the 37th Int Conf on Very Large Data Bases (VLDB). New York: ACM, 2011: 1470-1473

[13] Tran T, Pham T, Carneiro G, et al. A Bayesian data augmentation approach for learning deep models [C] //Proc of Advances in Neural Information Processing Systems (NIPS). Cambridge, MA: MIT Press, 2017: 2794-2803

[14] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks [C] // Proc of Advances in Neural Information Processing Systems (NIPS). Cambridge, MA: MIT Press, 2012: 1097-1105

[15] Prechelt L. Automatic early stopping using cross validation: Quantifying the criteria [J]. Neural Networks, 1998, 11(4): 761-767

[16] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting [J]. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958

[17] Loughrey J, Cunningham P. Using early-stopping to avoid overfitting in wrapper-based feature selection employing stochastic search [R]. Ireland: Trinity College Dublin, Department of Computer Science, 2005

[18] Nowozin S, Cseke B, Tomioka R. f-gan: Training generative neural samplers using variational divergence minimization [C] // Proc of Advances in Neural Information Processing Systems (NIPS). Cambridge, MA: MIT Press, 2016: 271-279

[19] Huszár F. How (not) to train your generative model: Scheduled sampling, likelihood, adversary? [J]. arXiv preprint arXiv:1511.05101, 2015

[20] Arjovsky M, Bottou L. Towards principled methods for training generative adversarial networks [J]. arXiv preprint arXiv:1701.04862, 2017

[21] Arjovsky M, Chintala S, Bottou L. Wasserstein gan [J]. arXiv preprint arXiv:1701.07875, 2017

[22] Mukkamala M C, Hein M. Variants of RMSProp and adagrad with logarithmic regret bounds [J]. arXiv preprint arXiv:1706.05507, 2017

[23] Tieleman T, Hinton G. Divide the gradient by a running average of its recent magnitude [J]. COURSERA: Neural Networks for Machine Learning, 2012, 4(2): 26-31

[24] Zhang Jiangwei, Tay Y C. Dscaler: Synthetically scaling a given relational database [C] //Proc of the 42nd Int Conf on Very Large Data Bases (VLDB). New York: ACM, 2016: 1671-1682

Song Kehui, born in 1994. PhD candidate. Her main research interests include database scaling, information retrieval and machine learning.

Zhang Ying, born in 1986. PhD, associate professor. Her main research interests include sentiment analysis, data mining, and information retrieval.

Zhang Jiangwei, born in 1990. PhD. His main research interests include database scaling, with particular focus on social network data.

Yuan Xiaojie, born in 1963. PhD, professor, PhD supervisor. Her main research interests include data management and data mining.