文章编号:1007－6735(2023)03－0235－09 DOI: 10.13255/j.cnki.jusst.20220307005

基于梯度惩罚生成对抗网络的过采样算法

陶家亮[1]， 魏国亮[2]， 宋 燕[3]， 窦 军[3]， 穆伟蒙[1]
（1. 上海理工大学 理学院，上海 200093；2. 上海理工大学 管理学院，上海 200093；
3. 上海理工大学 光电信息与计算机工程学院，上海 200093）

摘要： 在不平衡数据分类问题中，为了更注重学习原始样本的概率密度分布，提出基于梯度惩罚生成对抗网络的过采样算法(OGPG)。该算法首先引入生成对抗网络(GAN)，有效地学习原始数据的概率分布；其次，采用梯度惩罚对判别器输入项的梯度二范数进行约束，降低了 GAN 易出现的过拟合和梯度消失，合理地生成新样本。实验部分，在 14 个公开数据集上运用 k 近邻和决策树分类器对比其他过采样算法，在评价指标上均有显著提升，并利用 Wilcoxon 符号秩检验验证了该算法与对比算法在统计学上的差异。结果表明该算法具有良好的有效性和通用性。

关键词： 不平衡数据；过采样算法；概率密度分布；生成对抗网络；梯度惩罚
中图分类号： TP 181 **文献标志码：** A

Oversampling algorithm based on gradient penalty generative adversarial network

TAO Jialiang[1], WEI Guoliang[2], SONG Yan[3], DOU Jun[3], MU Weimeng[1]
(1. *College of Science, University of Shanghai for Science and Technology, Shanghai 200093, China*; 2. *Business School, University of Shanghai for Science and Technology, Shanghai 200093, China*; 3. *School of Optical−Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China*)

Abstract: In order to pay more attention to learning for probability density distribution of original samples in imbalanced data classification problem, an oversampling algorithm based on the gradient penalty generation adversarial network (OGPG) was proposed. Firstly, generation adversarial network (GAN) was adopted to effectively learn the probability density distribution of original data. Secondly, the gradient penalty was used to constrain the gradient two-norm of the input term of discriminator, which reduced the overfitting and gradient disappearance that appeared easily in GAN, so that the new samples were reasonably generated. In the experiment, the *k*-nearest neighbor and decision tree classifiers were adopted to compare the other oversampling algorithms, the evaluation indicators were significantly improved. The Wilcoxon signed-rank test was used to verify the statistical difference between this algorithm and the comparison algorithm. The results show that this algorithm has good effectiveness and generality.

收稿日期: 2022−03−07
基金项目: 国家自然科学基金资助项目 (61873169)；上海市“科技创新行动计划”国内科技合作项目 (20015801100)
第一作者: 陶家亮(1997−)，男，硕士研究生．研究方向：大数据分析．E-mail：2440039115@qq.com
通信作者: 魏国亮(1973−)，男，教授．研究方向：大数据分析、多智能体协同控制．E-mail：guoliang.wei@usst.edu.cn

Keywords: *imbalanced data*; *oversampling algorithm*; *probability density distribution*; *GAN*; *gradient penalty*

不平衡数据的分类问题在数据挖掘和机器学习领域中一直倍受关注。美国人工智能协会和国际机器学习会议分别就这个问题举行了研讨会。现实生活中，很多领域都会出现数据不平衡的问题，例如金融诈骗[1]、精准医疗[2]、故障诊断[3]、人脸识别[4-5]等。

数据不平衡[6]是指数据中某些类别的样本数量远比其他类别的多。通常情况下，少数类数据中包含更多重要的信息，是研究者重点关注对象。

目前处理不平衡数据分类的方法可以分为两大类：基于算法层面[7]和基于数据层面[8]。算法层面主要包括代价敏感学习[9]和集成学习[10]：代价敏感学习通过最小化贝叶斯风险确定代价函数，以最小化误分类代价为目标，但是误分类代价的先验信息是难以获得的；集成学习是将多个分类器的分类结果结合在一起，提高集成分类器的精度，进而关注少数类的重要性。但这两类算法没有改变数据分布。数据层面主要包括欠采样技术[11]、过采样技术[12]。数据层面的技术主要通过改变样本比例，例如欠采样技术主要是通过减少多数类样本，使得多数类样本和少数类样本趋于平衡，但随机地舍弃样本可能会丢失潜在的有用信息。随机过采样方法通过随机复制少数类样本，但是该方法只是简单的复制样本，增加了过拟合的风险。目前，过采样技术的应用较为广泛，因为该技术不仅保证了数据平衡，还没有损失原始数据的有效信息。

过采样技术的研究有很多，例如 Chawla 等[13]提出了合成少数类过采样技术 (synthetic minority oversampling technique, SMOTE)，该算法在少数类样本中与其近邻样本之间线性插值合成新样本，没有考虑少数类样本内部的数据分布情况。He 等[14]提出了自适应合成（adaptive synthetic, ADASYN）过采样方法，该算法通过样本点的学习难易程度给少数类样本赋予权值。此外，为了加强对边界样本的学习，边界自适应合成过采样技术[15](B-SMOTE1, B-SMOTE2）被提出。随着深度学习的高速发展，基于网络过采样的算法应运而生，Goodfellow 等[16]提出生成对抗网络（generative adversarial network, GAN）模型，通过生成器网络学习原始数据的分布。Douzas 等[17]提出利用条件生成对抗网络学习原始数据的分布，再对少数类进行过采样算法。何新林等[18]提出了基于隐变量后验生成对抗网络的过采样算法（latent posterior based GAN for oversampling,LGOS），该算法引入隐变量模型，降低了高斯噪声对生成样本的随机性影响。但 GAN 在训练过程易出现过拟合或梯度消失的风险，可以对损失函数施加惩罚项[19]，降低风险的发生。上述方法虽然在分类精度上有所提升，但没有充分考虑原始数据的分布，进而影响合成样本的安全性以及分类结果。

针对上述问题，本文提出了一种基于梯度惩罚生成对抗网络的过采样算法（oversampling algorithm based on the gradient penalty generation adversarial network，OGPG）。该算法引入生成对抗网络，通过网络的生成器模型有效地学习原始数据的概率密度分布；运用梯度损失模型对生成对抗网络判别器输入项的梯度二范数进行约束，降低过拟合和梯度消失的风险；在 14 个公共数据集上采用两个分类器与多种算法进行了对比实验，并利用 Wilcoxon 符号秩检验[20]验证了所提算法的有效性和通用性。

1 生成对抗网络模型及梯度惩罚模型

生成对抗网络（generative adversarial network，GAN）模型是一种无监督的生成模型，由生成器和判别器网络组成，能够有效地学习原始数据的概率密度分布。梯度惩罚模型是一种基于梯度损失的约束模型，降低了生成对抗网络出现过拟合和梯度消失的风险。

1.1 生成对抗网络模型

GAN 是 Goodfellow 等提出来的一种神经网络模型，也是一种无监督的生成模型。它由生成器网络和判别器网络两部分组成，网络模型结构如图 1 所示。GAN 也是一个相互博弈的对抗模型，是判别器和生成器之间的相互博弈。其中，生成器是通过对先验噪声的学习，学习原始数据的概率密度分布；判别器主要对输入数据进行判断，判断数据是原始数据或者是生成器网络生成的数

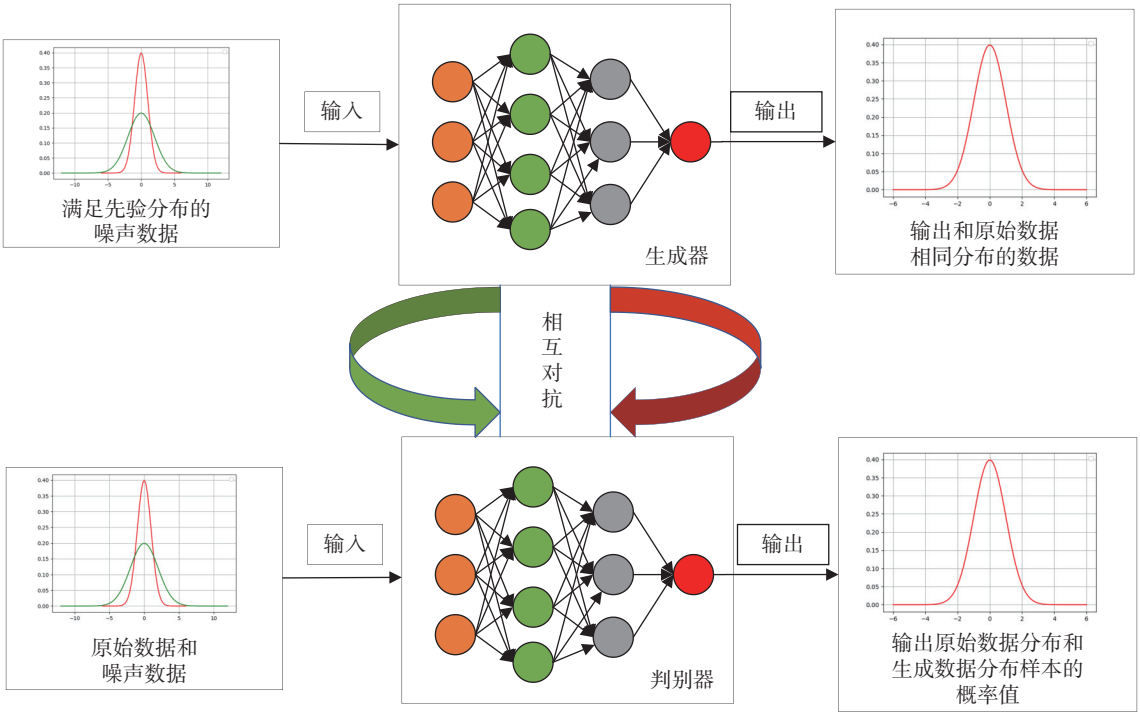


图 1　生成对抗网络图示

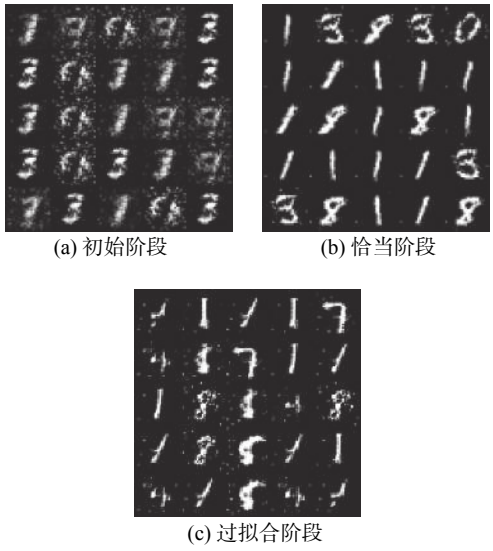
Fig.1　Graph of GAN

据，输出的是 0～1 之间的一个概率值。设噪声样本为 z，生成器通过映射将噪声样本转化为生成样本 $G(z)$。判别器输出 $D(x)$为 0～1 之间的概率值，可得其损失函数为

$$\min_G \max_D V(G,D)= E_{x\sim P_r}[\log(D(x))]+E_{z\sim P_z}[\log(1-D(G(z)))] \quad (1)$$

式中：E 表示期望值；P_r 表示真实样本 x 的概率密度分布；P_z 表示噪声样本 z 的概率密度分布。

对于 GAN 模型的训练阶段可以大致分为 3 个阶段，分别记为初始阶段、恰当阶段和过拟合阶段。为了能更清楚地解释上述现象，通过公开的 MNIST 手写数字体数据集进行了实验验证，结果见图 2。MNIST 数据集包含 60 000 个训练集样本和 10 000 个测试集样本，采用数据集的训练集样本对网络进行训练。初始阶段对应训练为 500 次；恰当阶段对应训练为 3 000 次；过拟合阶段对应训练为 8 000 次。



(a) 初始阶段　(b) 恰当阶段

(c) 过拟合阶段

图 2　MNIST 数据集实验结果

Fig. 2　Experimental results of MNIST dataset

1.2　梯度惩罚模型

梯度惩罚模型是 Gulrajani 等[21]提出来的针对 Wasserstein GAN 算法[22]存在生成样本的质量较差和模型不收敛等问题的约束惩罚算法模型。

对于该梯度惩罚模型，设P_r，P_g是紧凑度量空间的两个概率分布，f^*是可微的 L-利普希茨函数，处理下列优化问题：

$$\max_{\|f^*\|_L} E_{y\sim P_r}[f(y)]-E_{x\sim P_g}[f(x)] \quad (2)$$

设π是P_r，P_g的联合优化组合函数，定义距离度量 Wasserstein 距离为

$$W(P_r,P_g)=\inf_{\pi\in\Pi(P_r,P_g)}E_{(x,y)\sim\pi(P_r,P_g)}[\|x-y\|] \quad (3)$$

式中：y 为符合联合分布 π 的真实样本；$\Pi(P_r,P_g)$是联合分布$\pi(x,y)$的集合。由于f^*可微，则有

$$\pi(x=y)=0 \quad (4)$$

$$x_t = tx + (1-t)y, t \in [0,1] \tag{5}$$

$$P_{(x,y)\sim\pi}\left[\nabla f^*(x_t) - \frac{y - x_t}{\|y - x_t\|}\right] = 1 \tag{6}$$

即，对于所有的 L-利普希茨函数几乎都满足，若该函数可微则处处都有梯度，且梯度的范数值为1。根据上述理论知识，Ishaan 等研究者将梯度范数约束在不大于 1 的范围之内，提出如下新的约束惩罚：

$$L_{\mathrm{GP}} = \alpha E_{\bar{x}\sim P_{\bar{x}}}\left[(\|\nabla_{\bar{x}} D_w(\bar{x})\|_2 - 1)^2\right] \tag{7}$$

式中：L_{GP}表示梯度惩罚损失；$\bar{x}$表示训练样本；$\|\nabla_{\bar{x}} D_w(\bar{x})\|_2$表示 Wasserstein GAN 中判别器网络输入项梯度的二范数；α是梯度惩罚因子；w 是判别器网络的参数，即$D(\bar{x};w)$。

2 基于梯度惩罚生成对抗网络的过采样算法

由于传统的过采样算法没有充分考虑原始样本的概率密度分布，且易导致生成低质量的样本，因此本文引入生成对抗网络模型和梯度惩罚模型，提出了一种基于梯度惩罚生成对抗网络的过采样算法（OGPG）来解决上述问题。

在 OGPG 算法中，为防止少数类样本过少导致网络模型学习不到原始数据的有效信息，先对原始数据中的少类样本自适应生成部分样本。该算法主要包括 3 个步骤。

a. 去除噪声样本。

在数据预处理阶段，先处理原始数据中存在的噪声数据。对每个样本采用 k 近邻算法，计算样本点与其他样本点的距离，找到该样本点的 k 个最近邻样本点，如果该样本点的标签与 k 近邻中的所有样本点的标签不一致，则认定为噪声数据，并删除该样本点。

b. 合成部分少数类样本。

在步骤 (a) 的基础上，通过线性插值优先合成部分少数类样本数据，通过合成后的样本，学习样本的均值和方差，以便后续训练网络生成新的样本。

首先，设 T 为去噪后原始数据的总样本集合，T_{maj}为多数类样本集合，T_{min}为少数类样本集合，则有

$$T = T_{\mathrm{maj}} + T_{\mathrm{min}} \tag{8}$$

过采样所需要的生成的样本量

$$\tilde{T} = T_{\mathrm{maj}} - T_{\mathrm{min}} \tag{9}$$

接着，采用线性插值合成部分少数类样本，对于任意的T_{min}中的一个样本点x_i，运用欧氏距离度量，随机选取 k 近邻中的一个近邻样本x_j，通过线性插值合成样本$\tilde{x}$,

$$\tilde{x} = x_i + \mu(x_j - x_i) \tag{10}$$

式中，$\mu \in [0,1]$,通过线性插值合成的样本量集合记为T_{syn}。通过合成少数类样本后得到新的少数类样本集合记为$T_{\mathrm{new_min}}$。其中，

$$T_{\mathrm{syn}} = \frac{\tilde{T}}{2} \tag{11}$$

$$T_{\mathrm{new\_min}} = T_{\mathrm{min}} + T_{\mathrm{syn}} \tag{12}$$

c. 生成新样本。

结合生成对抗网络模型和梯度惩罚模型优良性质，针对过采样问题提出了改进后的损失函数为

$$\begin{aligned} L_1 = &\min_G \max_D E_{x\sim P_{\mathrm{r}}}[\log(D(x))] + \\ &E_{x\sim P_{\mathrm{g}}}[\log(1 - D(x))] + \alpha E_{\bar{x}\sim P_{\bar{x}}}\left[(\|\nabla_{\bar{x}} D(\bar{x})\|_2 - 1)^2\right] \end{aligned} \tag{13}$$

式中，$P_{\bar{x}}$表示真实数据分布和生成数据分布采样的线性均匀采样分布，即 $\bar{x} = \beta x_{\mathrm{r}} + (1-\beta)x_{\mathrm{g}}$, $\beta \in (0,1)$。

通过步骤 (a) 的去除噪声和步骤 (b) 合成部分少数类样本之后，采用梯度惩罚生成对抗网络算法生成新样本。

首先，把合成的新的少数类样本记为新少数类样本，即$T_{\mathrm{new_min}}$。通过计算得到该样本的均值和方差，分别记为μ和σ^2。对于噪声样本 z，假设满足

$$z \sim P_z \sim N(\mu, \sigma^2) \tag{14}$$

噪声数据通过映射将数据转化为生成样本

$$x = G(z) \tag{15}$$

接着，将噪声样本和新少数类样本分别用生成器网络和判别器网络进行迭代，计算各个网络及梯度惩罚的损失，由式 (12) 得到判别器损失 L_{D}、生成器损失 L_{G} 和梯度惩罚损失 L_{GP}，分别为

$$L_{\mathrm{D}} = -E_{x\sim P_{\mathrm{r}}}[\log D(x)] - E_{x\sim P_{\mathrm{g}}}[\log(1 - D(x))] \tag{16}$$

$$L_{\mathrm{G}} = -E_{x\sim P_{\mathrm{g}}}[\log(1 - D(x))] \tag{17}$$

$$L_{\mathrm{GP}} = \alpha E_{\bar{x}\sim P_{\bar{x}}}\left[(\|\nabla_{\bar{x}} D(\bar{x})\|_2 - 1)^2\right] \tag{18}$$

式中： x为训练样本；$\|\nabla_x D(x)\|_2$为求该样本的梯度的二范数。

再设置判别器网络和生成器网络的收敛阈值，在达到阈值之后停止迭代，实验设置循环迭代阈值为 3 000 次。最后，通过网络收敛时生成器生成的样本即为新样本，通过梯度惩罚的生成对抗网络模型生成的样本集合记为T_{gen}。

根据上述对于 OGPG 算法步骤的描述，给出算法的合成样本示意图，见图 3。

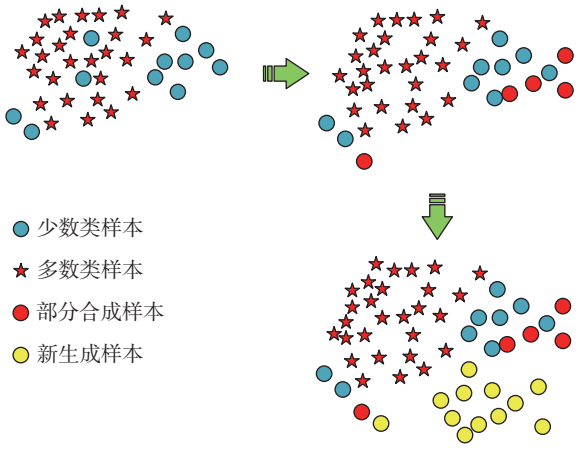


图 3　合成样本示意图

Fig.3　Schematic diagram of synthetic sample

3　实验结果及分析

3.1　数据集

为了验证 OGPG 算法的有效性，实验从 UCI 机器学习库中挑选了 14 组二类不平衡数据集，其样本量、特征数以及不平衡率（imbalanced ratio，IR）都不相同。表 1 是所选取的数据集的详细信息：

3.2　评价指标

在处理不平衡数据的分类问题的时候，分类器的超平面会向少数类样本偏移，因此精确率不适合作为评价指标。实验采用F_{m}和G_{m}作为评价指标[23]。其中 F_{m}表示单一类别精确率和召回率的均衡指标，G_{m}表示召回两个类别数据的综合表现指标。F_{m}和G_{m}的计算式如下：

$$P=\frac{TP}{TP+FP},\ R=\frac{TP}{TP+FN} \tag{19}$$

$$S=\frac{TN}{TN+FP} \tag{20}$$

$$F_{\text{m}}=\frac{2PR}{P+R} \tag{21}$$

表 1　实验数据集信息

Tab.1　Details of datasets

数据集	样本数	特征数	多数类	少数类	IR
wisconsin	683	9	444	239	1.857 7
yeast1	1 484	10	1 055	429	2.459 2
pageblocks0	5 472	10	4 913	559	8.788 9
abalone9_18	731	8	689	42	16.404 0
cargood	1 728	6	1 659	69	24.043
winequality_red4	1 599	11	1 546	53	29.169
abalone3vs11	502	8	487	15	32.466
ecoli0137vs26	281	7	274	7	39.143
phoneme	5 404	5	3 818	1 586	2.41
satimage	4 435	36	3 956	479	8.26
pen	10 992	16	9 937	1 055	9.42
wine	6 497	11	5 617	880	6.38
letter	20 000	16	18 445	1 555	11.86
avila	10 430	10	9 335	1095	8.53

$$G_{\text{m}}=\sqrt{RS} \tag{22}$$

式中：TP 表示将正例样本预测为正例；FP 表示将正例样本预测为反例；FN 表示将反例样本预测为正例；TN 表示将反例样本预测为反例；P 为查准率；R 为召回率；S 为特异性。

3.3　实验分析

为了验证 OGPG 算法的优越性，首先通过前 8 组数据集对比了 SMOTE, ADASYN，B-SMOTE，CBSO[24] 传统过采样算法。其次通过后 4 组数据集对比了采用 GAN 的 LGOS 算法。此外，在对比传统算法中，采用 k 近邻分类器和决策树分类器随机选取 70% 的数据作为测试集，剩余 30% 的数据作为测试集，每个数据集取 5 次实验结果的平均值作为报告结果。在对比 LGOS 算法中采用决策树分类器选取 80% 的数据作为测试集，剩余 20% 的数据作为测试集，每个数据集取 10 次实验结果的平均值作为报告结果。粗体表示的是实验的最优值。通过上述实验验证本算法的有效性和泛化能力。所有实验都是在 2.80 GHz CPU、16.0 GB 内存的电脑上运行的，软件环境是 Python3.7。

从表 2 和表 3 的结果可以看出，无论是 k 近邻分类器还是决策树分类器，OGPG 算法在 F_{m}，G_{m}上均获得了明显提升。在 F_{m}指标下，8 个数据集中都表现较好；在 G_{m}指标下，8 个数据集中 7 个表现相对较好。通过对表 2、表 3 对各指标的

分析，可以发现算法在 G_m 指标下 abalone3vs11 数据集上表现相对没有优势。该数据集在 CBSO 算法上表现相对较好，之所以出现该现象，是因为数据集中存在边界较难学习的样本，OGPG 算法较难学习到该样本的有效信息，导致评价指标相对较低。但是从结果上看仍然非常接近最优指标，充分说明了 OGPG 算法的有效性。通过上述对表 2 和表 3 的结果分析，验证了 OGPG 算法的有效性。

为了验证 OGPG 算法的稳定性，实验绘制了数据集在 F_m 指标和 G_m 指标下的箱线图，分别见图 4 和图 5。箱线图包括一个矩形箱体和上下两条线，

表 2 基于 *k* 近邻和决策树分类器的 F_m 指标

Tab.2 F_m based on *k*-nearest neighbor and decision tree classification

数据集	分类器	SMOTE	ADASYN	B-SMOTE	CBSO	OGPG
yeast1	*k*近邻	0.993 8	0.996 9	0.993 7	0.996 5	**0.997 2**
	决策树	**0.998 9**	0.997 5	0.998 4	0.998 2	0.997 5
ecoli0137vs26	*k*近邻	0.988 7	**0.994 2**	0.988 5	0.988 5	**0.994 2**
	决策树	0.988 8	0.988 3	0.988 5	0.994 2	**0.998 8**
abalone9_18	*k*近邻	0.989 1	0.974 2	0.987 2	0.988 5	**0.995 3**
	决策树	0.963 2	0.962 9	**0.978 2**	0.973 7	0.978 1
abalone3vs11	*k*近邻	0.939 5	0.939 3	0.948 3	0.961 6	**1**
	决策树	0.925 2	0.949 9	0.946 4	0.973 2	**1**
pageblock0	*k*近邻	0.987 3	0.983 3	**0.988 7**	0.981 9	0.987 3
	决策树	0.986 2	0.986 4	0.987 9	0.985 8	**0.988 8**
winequality_red4	*k*近邻	0.989 6	0.988 1	0.976 6	0.986 9	**0.991 5**
	决策树	0.984 8	0.974 8	0.982 7	0.975 6	**0.986 4**
car_good	*k*近邻	0.895 1	0.892 1	0.902 5	0.885 9	**0.902 7**
	决策树	0.913 3	0.917 1	0.892 6	0.903 8	**0.922 1**
wisconsin	*k*近邻	0.960 2	0.963 5	0.952 4	0.956 5	**0.973 1**
	决策树	0.948 8	0.951 9	0.932 5	0.960 4	**0.976 4**

表 3 基于 *k* 近邻和决策树分类器的 G_m 指标

Tab.3 G_m based on *k*-nearest neighbor and decision tree classification

数据集	分类器	SMOTE	ADASYN	B-SMOTE	CBSO	OGPG
yeast1	*k*近邻	0.993 6	0.986 7	0.993 6	0.992 1	**0.996 8**
	决策树	0.993 6	0.996 7	0.993 6	0.996 8	**0.997 5**
ecoli0137vs26	*k*近邻	0.986 9	0.993 6	0.987 8	0.987 2	**0.993 7**
	决策树	0.986 9	0.993 7	0.987 8	0.987 2	**0.998 7**
abalone9_18	*k*近邻	0.986 2	0.987 9	0.983 2	0.987 2	**0.993 4**
	决策树	0.986 5	**0.990 4**	0.983 2	0.987 2	0.977 1
abalone3vs11	*k*近邻	0.935 5	0.924 7	0.943 8	**0.962 3**	0.949 9
	决策树	0.935 2	0.926 8	0.943 8	**0.973 7**	0.954 7
pageblock0	*k*近邻	0.987 4	0.983 3	**0.988 9**	0.981 6	0.987 1
	决策树	0.986 3	0.986 5	0.988 1	0.985 5	**0.989 4**
winequality_red4	*k*近邻	0.988 9	0.987 7	0.978 1	0.987 2	**0.991 3**
	决策树	0.978 6	0.973 1	0.983 8	0.975 7	**0.986 1**
car_good	*k*近邻	0.878 7	0.878 1	0.890 8	0.870 6	**0.897 7**
	决策树	0.915 8	0.919 2	0.895 8	0.904 9	**0.924 9**
wisconsin	*k*近邻	0.957 9	0.962 2	0.956 8	0.954 4	**0.974 5**
	决策树	0.946 9	0.952 9	0.937 1	0.959 2	**0.977 5**

箱体中间的线为中位线，上限和下限分别为上四分位数和下四分位数，箱子的宽度显示数据的波动程度，箱体的上下方各有一条线是数据的最大值和最小值，超出最大最小值线的数据为异常数据。从图 4 和图 5 中可以看出，OGPG算法的数据波动性相对较小，数据的中值、上下四分位数与其他算法相比要更加稳定，且数值也优于其他算法，这说明了 OGPG 算法稳定性较好。

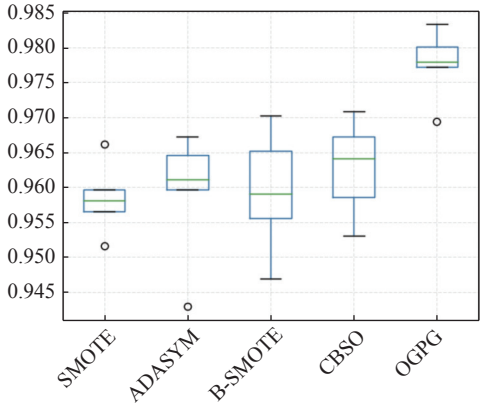


图 4　基于 F_m 指标的箱线图

Fig.4　Boxplot based on F_m

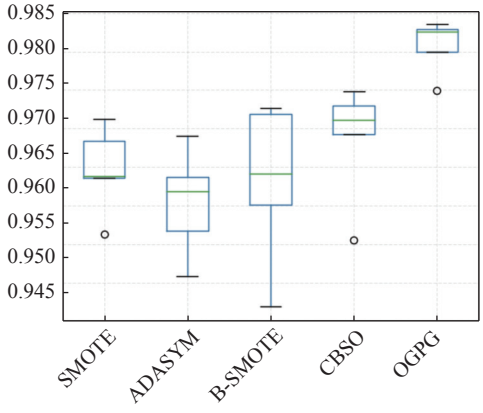


图 5　基于 G_m 指标的箱线图

Fig.5　Boxplot based on G_m

为了验证 OGPG 算法在统计学上是否具有显著性，本文采用 Wilcoxon 符号秩检验来评估所提算法和其他对比算法之间的显著性差异。表 4～表 7 是 Wilcoxon 符号秩检验的结果，其中 *R*+表示所提算法的秩和，*R*–表示对比算法的秩和，置信度是 95%，*p* 为 0.05。在 *k* 近邻分类器下，可以看到，都是拒绝原假设；在决策树分类器下，在对比算法 ADASYN、CBSO 在 G_m 指标下是接受原假设，其余都是拒绝原假设，说明 OGPG 算法相对于其他算法具有较显著的差异性。结合表 2、表 3 在各指标的综合表现情况，说明 OGPG 算法相对于传统算法有显著的有效性。

表 4　基于 F_m 及 *k* 近邻分类器的 Wilcoxon 检验表

Tab.4　Wilcoxon test on *k*-nearest neighbor classifier of F_m

对比方法	*R*+	*R*–	*p*
OGPG vs SMOTE	28	0	0.014
OGPG vs ADASYN	28	0	0.014
OGPG vs B-SMOTE	34	2	0.023
OGPG vs CBSO	36	0	0.008

表 5　基于 G_m 及 *k* 近邻分类器的 Wilcoxon 检验表

Tab.5　Wilcoxon test on *k*-nearest neighbor classifier of G_m

对比方法	*R*+	*R*–	*p*
OGPG vs SMOTE	35	1	0.015
OGPG vs ADASYN	36	4	0.008
OGPG vs B-SMOTE	35	1	0.015
OGPG vs CBSO	30	6	0.109

表 6　基于 F_m 及决策树分类器的 Wilcoxon 检验表

Tab.6　Wilcoxon test on decision tree classifier of F_m

对比方法	*R*+	*R*–	*p*
OGPG vs SMOTE	35	1	0.016
OGPG vs ADASYN	28	0	0.017
OGPG vs B-SMOTE	33	3	0.047
OGPG vs CBSO	35	1	0.016

表 7　基于 G_m 及决策树分类器的 Wilcoxon 检验表

Tab.7　Wilcoxon test on decision tree classifier of G_m

对比方法	*R*+	*R*–	*p*
OGPG vs SMOTE	31	5	0.041
OGPG vs ADASYN	30	6	0.053
OGPG vs B-SMOTE	32	4	0.029
OGPG vs CBSO	26	10	0.153

为了全面验证算法的有效性，实验还对比了文献 [18] 的 LGOS 算法，即采用 GAN 的过采样算法，如表 8 所示。从表 8 的结果可以看出，在决策树分类器下，无论是 F_m 还是 G_m 指标，该算法均有较为明显的提升。除此之外，在前 8 组数据集中，样本量相对较少，在对比传统算法中有显著提升；在后 6 组数据集中，数据样本量相对较多，在对比算法中同样有着较为明显的提升，说

表 8 基于决策树分类器的对比评价

Tab.8 Evaluation based on decision tree classifier

数据集	评价指标	LGOS	OGPG
phoneme	F_m	0.758 6	**0.938 4**
	G_m	0.848 6	**0.945 9**
satimage	F_m	0.946 1	**0.971 9**
	G_m	0.977 8	**0.983 3**
pen	F_m	0.968 1	**0.998 9**
	G_m	0.990 2	**0.999 2**
wine	F_m	0.596 6	**0.993 6**
	G_m	0.789 7	**0.997 8**
letter	F_m	0.893 6	**0.985 4**
	G_m	0.950 1	**0.983 5**
avila	F_m	0.951 1	**0.999 3**
	G_m	0.848 6	**0.999 7**

明了算法的有效性。

OGPG 算法和 LGOS 算法之间的显著性差异见表 9。可以看出，在置信度为 95% 的情况下，即 p 不大于 0.05 的情况下，均拒绝原假设。说明 OGPG 算法相对于 LGOS 算法具有显著的差异性。通过该部分实验也说明了 OGPG 算法具有显著的有效性。

表 9 Wilcoxon 符号秩检验表

Tab.9 The table of Wilcoxon signed rank test

对比方法	评价指标	$R+$	$R-$	p
OGPG vs LGOS	F_m	21	0	0.028
OGPG vs LGOS	G_m	21	0	0.026

4 结束语

针对不平衡数据分类问题，传统的过采样算法没有充分考虑原始数据的概率密度分布，从而导致生成的样本不具有较强的安全性。通过引入生成对抗网络以及梯度惩罚模型，提出了一种基于梯度惩罚生成对抗网络的过采样算法。在该算法中，首先引入生成对抗网络，通过生成器网络有效地学习原始数据的概率密度；其次，由于生成对抗网络易出现过拟合或梯度消失等现象，因此采用梯度惩罚来对判别器网络输入项的梯度二范数进行约束，从而有效地降低了该情况的发生，使得生成器既能有效学习数据的概率密度分布又能合理地生成新样本；最后，在 14 个公共数据集上采用两个分类器与多种算法进行了对比实验，并利用 Wilcoxon 符号秩检验验证了所提算法的有效性和通用性。当然，该算法也有一定的缺点，在时间复杂度上，因为算法引入了深度学习网络，所以时间复杂度上较高，这也是后续将要努力的方向。

参考文献：

[1] FIORE U, DE SANTIS A, PERLA F, et al. Using generative adversarial networks for improving classification effectiveness in credit card fraud detection[J]. Information Sciences, 2019, 479: 448–455.

[2] FOTOUHI S, ASADI S, KATTAN M W. A comprehensive data level analysis for cancer diagnosis on imbalanced data[J]. Journal of Biomedical Informatics, 2019, 90: 103089.

[3] MENA L J, GONZALEZ J A. Machine learning for imbalanced datasets: application in medical diagnostic[C]//Proceedings of the Nineteenth International Florida Artificial Intelligence Research Society Conference. Melbourne Beach: AAAI Press, 2006: 574–579.

[4] 武文娟, 李勇. Emfacenet: 一种轻量级人脸识别的卷积神经网络 [J/OL]. 小型微型计算机系统, 2021: 1–6. (2021-12-17). http://kns.cnki.net/kcms/detail/21.1106.tp.20211214.1436.004.html.

[5] 周建含, 李英梅, 李文昊. 一种改进的半监督集成软件缺陷预测方法 [J]. 小型微型计算机系统, 2021, 42(10): 2196–2202.

[6] ZHANG H L, LIU G S, PAN L, et al. GEV regression with convex loss applied to imbalanced binary classification[C]//2016 IEEE First International Conference on Data Science in Cyberspace (DSC). Changsha: IEEE, 2016: 532–537.

[7] JING X Y, ZHANG X Y, ZHU X K, et al. Multiset feature learning for highly imbalanced data classification[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(1): 139–156.

[8] ZHENG Z Y, CAI Y P, LI Y. Oversampling method for imbalanced classification[J]. Computing and Informatics, 2015, 34(5): 1017–1037.

[9] CASTRO C L, BRAGA A P. Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data[J]. IEEE Transactions on Neural Networks and Learning Systems, 2013, 24(6): 888–899.

[10] WANG C, DENG C Y, YU Z L, et al. Adaptive ensemble of classifiers with regularization for imbalanced data

classification[J]. Information Fusion, 2021, 69: 81–102.
[11] 周传华, 朱俊杰, 徐文倩, 等. 基于聚类欠采样的集成分类算法 [J]. 计算机与现代化, 2021(11): 72–76.
[12] 陈刚, 郭晓梅. 基于时间序列模型的非平衡数据的过采样算法 [J]. 信息与控制, 2021, 50(5): 522–530.
[13] CHAWLA N V, BOWYER K W, HALL L O, et al. SMOTE: synthetic minority over-sampling technique[J]. Journal of Artificial Intelligence Research, 2002, 16: 321–357.
[14] HE H B, BAI Y, GARCIA E A, et al. ADASYN: Adaptive synthetic sampling approach for imbalanced learning[C]//2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). Hong Kong, China: IEEE, 2008: 1322–1328.
[15] HAN H, WANG W Y, MAO B H. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning[C]//International Conference on Intelligent Computing. Berlin, Heidelberg: Springer, 2005: 878–887.
[16] GOODFELLOW I J, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial nets[C]//Proceedings of the 27th International Conference on Neural Information Processing Systems. Montreal: MIT Press, 2014: 2672–2680.
[17] DOUZAS G, BACAO F. Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE[J]. Information Sciences, 2019, 501: 118–135.
[18] 何新林, 戚宗锋, 李建勋. 基于隐变量后验生成对抗网络的不平衡学习 [J]. 上海交通大学学报, 2021, 55(5): 557–565.
[19] LUO X, CHANG X H, BAN X J. Regression and classification using extreme learning machine based on L_1-norm and L_2-norm[J]. Neurocomputing, 2016, 174: 179–186.
[20] CUZICK J. A Wilcoxon - type test for trend[J]. Statistics in Medicine, 1985, 4(1): 87–90.
[21] GULRAJANI I, AHMED F, ARJOVSKY M, et al. Improved training of Wasserstein GANs[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: Curran Associates Inc. , 2017: 5769–5779.
[22] ADLER J, LUNZ S. Banach Wasserstein GAN[C]//Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montréal: Curran Associates Inc. , 2018: 6755–6764.
[23] HE H B, GARCIA E A. Learning from imbalanced data[J]. IEEE Transactions on Knowledge and Data Engineering, 2009, 21(9): 1263–1284.
[24] YU Y, GAO S C, CHENG S, et al. CBSO: a memetic brain storm optimization with chaotic local search[J]. Memetic Computing, 2018, 10(4): 353–367.

（编辑：董　伟）