第51卷第5期 南京航空航天大学学报 Vol. 51 No. 5
2019年10月 Journal of Nanjing University of Aeronautics & Astronautics Oct. 2019

DOI:10.16356/j.1005-2615.2019.05.005

基于欠采样的零阶优化算法

鲁淑霞 张罗幻 蔡莲香

(1.河北大学数学与信息科学学院,河北省机器学习与计算机智能重点实验室,保定,071002)

摘要: 非平衡学习吸引了许多研究者的关注。一般情况下,少数类是更值得关注的,并且其误分类代价要远高于多数类。由于非平衡数据分布的非均衡性,标准的分类算法将难以适用。为了解决非平衡数据分类问题,给出了基于欠采样的零阶优化算法。首先,为了降低数据非平衡分布的影响,针对不同非平衡比的数据集给出了不同的两种采样策略。然后,采用了一种引入间隔均值项的支持向量机(Support vector machine, SVM)优化模型进行分类,并使用带有方差减小的零阶随机梯度下降算法进行求解,提高了算法的精度。在非平衡数据上进行了对比实验,实验证明提出的方法有效提高了非平衡数据的分类效果。

关键词: 欠采样;零阶优化;支持向量机;非平衡数据集;方差减小

中图分类号: TP391 **文献标志码:** A **文章编号:** 1005-2615(2019)05-0609-09

Zeroth Order Optimization Algorithm Based on Undersampling

LU Shuxia, ZHANG Luohuan, CAI Lianxiang

(College of Mathematics and Information Science, Hebei University, Hebei Province Key Laboratory of Machine Learning and Computational Intelligence, Baoding, 071002, China)

Abstract: In recent years, imbalanced learning has attracted the attention of many researchers. In general, minority classes are more noteworthy, and the cost of misclassification is much higher than that of majority classes. Because of the imbalanced distribution of imbalanced data, the standard classification algorithms will be difficult to apply. In order to solve the problem of imbalanced data classification, a zeroth-order optimization algorithm based on under-sampling is presented. Firstly, in order to reduce the influence of imbalanced data distribution, two different sampling strategies are adopted for data sets with different imbalanced ratios. Then, an SVM (Support vector machine) model with margin mean term is used for classification, and a zeroth-order stochastic gradient descent algorithm with reduced variance is used to solve the problem. At the same time, the accuracy of the algorithm is improved. A comparative experiment is carried out on imbalanced data, and the experimental results show that the proposed method effectively improves the classification effect of imbalanced data.

Key words: undersampling; zeroth order optimization; support vector machine (SVM); imbalanced data sets; variance reduction

传统的分类算法均假定数据集中的不同类样本的数量都是大致相同的。然而,在许多实际情况下,由于一些类的样本数量远大于其他类,这就导致了样本分布的不平衡,也给标准的分类算法带来了许多挑战。从数据挖掘的角度看,通常少数类更为重要,因为尽管少数类样本个数很少,但它却可能带有更重要的信息。例如,在金融欺诈检测中,如果未能在海量交易数据中识别出欺诈交易,则将

基金项目: 国家自然科学基金(61672205)资助项目。

收稿日期: 2019-03-31;**修订日期:** 2019-07-30

通信作者: 鲁淑霞,女,教授,E-mail:cmclusx@126.com。

引用格式: 鲁淑霞,张罗幻,蔡莲香.基于欠采样的零阶优化算法[J].南京航空航天大学学报,2019,51(5):609-617. LU Shuxia, ZHANG Luohuan, CAI Lianxiang. Zeroth Order Optimization Algorithm Based on Undersampling[J]. Journal of Nanjing University of Aeronautics & Astronautics,2019,51(5):609-617.

导致难以估量的损失。因此,采用合适的方法和策略,降低数据不同类别的非平衡比,对提高算法的分类性能至关重要。

目前,处理非平衡数据分类问题的方法主要分为3大类:数据层面、算法层面以及混合的方法。数据层面的方法,主要是通过对多数类样本进行欠采样[1]以及对少数类样本进行过采样[2]或者结合两种方法[3]对数据进行预处理,以达到数据分布相对平衡的效果。对于非平衡数据集,算法层面的方法主要是对现有的算法进行改进,以减轻分类器受多数类样本的影响。常见的非平衡数据分类的算法有:对不同类别的样例赋予不同惩罚参数的代价敏感学习[4]、基于专家决策的主动学习[5]以及主要用于极端事件检测中的单类学习[6]等。集成的方法是处理不均衡问题的混合方法中的一大类,持续受到了许多研究者的关注。文献[7]提出了一种基于欠采样和Real Adaboost组合的用于非平衡数据分类的新框架。在文献[8]中介绍了一种提升近邻欠采样支持向量机,并且还给出了一种核距离预计算技术。文献[9]提出了一种以决策树和神经网络作为基分类器的套袋与采样方法相结合的算法。

随着大数据时代的到来,生成新样本点的过采样方法,增加了原始数据集的数量,导致数据的处理时间和存储空间的消耗大大增加。例如,最简单的随机过采样[10]算法,它通过随机复制少数类样本生成新的样例。这种方法随机性较大,极易导致过拟合。另一种在少数类样本的最近邻样本点间进行线性插值的合成少数类过采样方法(Synthetic minority oversampling technique,SMOTE)[11],虽然一定程度上可以降低过拟合的风险,但并没有考虑到近邻样本的分布特点,易于生成噪声点。由此,为了不增加计算负担,欠采样的方法更适用于非平衡数据分类,尤其适用于非平衡大数据集。

随机欠采样[12]是一种常用的欠采样方法,它对多数类样例进行随机抽取,十分简单易用。但是,在这种方法选择样例时,由于其随机性较大,易导致具有重要信息样例的丢失,算法性能不佳。Qi等[13]提出了一种加权欠采样方法(Weighted under-sampling,WU),根据权重的大小进行采样,减小了采样过程中的随机性,但却不适用于高非平衡比数据集。基于WU算法的思想,给出了一种针对不同非平衡比数据集的欠采样方法。这种方法通过计算样本点到分类超平面的几何距离进行加权欠采样,对距离近的样本点赋予较大的权重,较远的样本点赋予了较小的权重。并且,它根据数据集非平衡比的大小分为两种采样方法:对于非平衡比低于10的数据集,更多地保留了数据的原始分布信息;而对于高于10的数据集,则考虑了采集具有重要信息的样例。

如何建立高效的分类模型,一直是非平衡学习领域的热点问题。在本文中,使用了一种引入间隔均值项的支持向量机模型。1995年,Vapnik等[14]首次提出了SVM的方法,它的核心思想在于最大化数据集的最小间隔,并没有考虑到数据分布的影响,导致对于非平衡数据分类效果较差,间隔分布对最终结果的影响至关重要[15]。因此,文中引入了间隔均值项SVM模型,不仅保持了SVM模型高精度的特点,亦进一步提高了模型的泛化能力。

Pegasos[16]算法是一种常用的随机优化算法,它在对优化问题的求解过程中使用单个样本梯度代替全梯度,有效节省了计算量,但却导致了较大方差,阻碍了收敛速度;此外,对于一些复杂的优化模型,难以求导或者无法进行求导,用该方法无法进行求解。针对上述不足,本文将使用函数值的梯度估计来近似梯度的零阶优化算法[17-18]与随机方差减小算法(Stochastic variance reduced gradient,SVRG)[19]两种方法的思想进行了结合,给出了一种带有方差减小的零阶优化方法(Zeroth order optimization method with variance reduction,ZOVR)。随着大数据平台的发展,文献[20]提出了一种异步环境下的带有方差减小的零阶随机方法,有效减小了系统开销。

对于传统的SVM分类算法在非平衡数据分类方面产生的不足,给出一种基于欠采样和减小方差的零阶优化算法(Zeroth order optimization method based on under sampling and variance reduced,U-ZOVR)。这种算法不仅减小了计算量,而且提高了算法的精度。

1 基于几何距离加权的欠采样方法

随机欠采样算法在采样过程中,对所有的样本赋予相同的采样率,这并不适用于非平衡数据集。并且,对于不同非平衡比的数据集,数据分布差异较大。因此,给出了基于几何距离加权的欠采样方法。根据非平衡比的大小,分为两种采样方法:非平衡比低于10的欠采样算法1(Under-sampling 1,U1)以及非平衡比高于10的欠采样算法2(Under-sampling 2,U2)。

文中的两种欠采样方法,是根据多数类样本点到SVM超平面的几何距离,对多数类样本点进行加权欠采样,距离越近,权值越大,即采样率越高。假设训练集为$\boldsymbol{S}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_n,y_n)\}$,其

中 $x_i \in \mathbf{R}^d$，类标 $y_i \in \{-1,1\}$，d 为样本维数，$i=1,2,\cdots,n$。多数类（负类）样本个数为 n_1，少数类（正类）样本个数为 n_2，标准的SVM定义为

$$\min (1/2)\|\boldsymbol{w}\|^2+\lambda_2\sum_{i=1}^{n}\xi_i \tag{1}$$

$$\text{s.t.}\quad y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)\geqslant 1-\xi_i,\xi_i\geqslant 0\quad i=1,2,\cdots,n$$

式中：λ_2 是惩罚参数，ξ_i 为松弛变量。由此可知，多数类样本点到SVM超平面的几何距离 γ 的计算公式为

$$\gamma=\left|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}+b\right|/\|\boldsymbol{w}\| \tag{2}$$

当得到所有多数类样本点到超平面的几何距离后，对于非平衡比低于10的U1算法，则将所有多数类样本分为 m 个平行子区域，每个子区域的直径 δ 等于样本点到超平面最大间隔距离与最小间隔距离的差值除以子区域的个数，计算公式为

$$\delta=(\gamma_{\max}-\gamma_{\min})/m \tag{3}$$

不同子区域内样本的权重大小不同，距离超平面越近的子区域内的样本被赋予较大权重，较远的子区域内的样本则权重较小，每个单独子区域内样本的权重相同。对于多数类样本 $\boldsymbol{x}_j(j=1,2,\cdots,n_1)$，$n_1$ 为多数类样本个数，样本 $\boldsymbol{x}_j$ 的权重，即采样率 ρ_{x_j} 的计算公式为

$$\rho_{x_j}=1-(\lceil\gamma_j/\delta\rceil-1)/m \tag{4}$$

非平衡比低于10的欠采样算法U1的伪代码如下。

算法1　U1算法

输入：数据集 $\boldsymbol{S}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_n,y_n)\}$

输出：采样集 $\boldsymbol{S}'=\{(\boldsymbol{x}'_1,y'_1),(\boldsymbol{x}'_2,y'_2),\cdots,(\boldsymbol{x}'_s,y'_s)\}$

(1) 初始化：所有样本的个数 n，多数类样本个数 n_1，子区域个数 m

(2) for $j=1,2,\cdots,n_1$

(3) 通过式(2)，计算多数类样本 x_j 到超平面的距离 γ_j

(4) end

(5) 利用式(3)，计算子区域的直径 δ

(6) for $j=1,2,\cdots,n_1$

(7) 根据计算所得的 γ_j，将多数类划分为 m 个子区域，用式(4)计算每个子区域内样本的采样率 ρ_{x_j}

(8) end

(9) 根据采样率进行采样，输出结合所有少数类样本的采样集 $\boldsymbol{S}'$，$\boldsymbol{S}'$ 中共包含样本 s 个

U1算法对所有的多数类样本进行了采样，最大化的保留了原始数据集的分布信息。但是，对于非平衡比较高的数据集却难以适用。这是由于，当数据的非平衡比较高时，远离超平面的样例对于决策超平面的贡献较小甚至没有贡献。因此，对高非平衡比数据集的所有多数类采样是不恰当的，也增加了计算开销。为了解决上述问题，用于高非平衡比数据集的U2算法被提出。该方法在采样前，先根据计算得出的几何距离对多数类样本进行了排序，其次截取距离超平面较近的样本，保存到集合 $\boldsymbol{S}_1$ 中，最后进行采样。为了较好的保留数据集的原始分布信息，在集合 $\boldsymbol{S}_1$ 中保存了数量为少数类样本10倍的多数类样本，即定义 $\boldsymbol{S}_1$ 中样本个数为 $n_3=10\times n_2$。所以算法U2在不影响最优决策超平面生成的同时，大量减小了计算开销。U2算法的伪代码如下。

算法2　U2算法

输入：数据集 $\boldsymbol{S}=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_n,y_n)\}$；

输出：采样集 $\boldsymbol{S}'=\{(\boldsymbol{x}'_1,y'_1),(\boldsymbol{x}'_2,y'_2),\cdots,(\boldsymbol{x}'_s,y'_s)\}$；

(1) 初始化：所有样本的个数 n，多数类样本个数 n_1，少数类样本个数 n_2，定义 $n_3=10\times n_2$，集合 $\boldsymbol{S}_1$

(2) for $j=1,2,\cdots,n_1$

(3) 通过式(2)，计算多数类样本 x_j 到超平面的距离 γ_j

(4) end

(5) 根据 γ 值对多数类样本进行降序排序，保留排在 n_3 的样本到集合 $\boldsymbol{S}_1$ 中，并根据 $\boldsymbol{S}_1$ 中样本对应的 $\gamma_{\max}$ 以及 $\gamma_{\min}$，计算子区域直径 δ

(6) for $j=1,2,\cdots,n_3$

(7) 根据计算所得的 γ_j，将多数类划分为 m 个子区域，用式(4)计算每个子区域内样本的采样率 ρ_{x_j}

(8) end

(9) 根据采样率进行采样，输出结合所有少数类样本的采样集 $\boldsymbol{S}'$，$\boldsymbol{S}'$ 中共包含样本 s 个

SVM对数据集进行分类时，其超平面的确定仅取决于支持向量，算法U2结合了SVM的这种思想，并考虑了非平衡数据集分布不均衡的特点，仅对数量为少数类10倍的多数类样本进行采样。在不影响分类结果的同时，尽可能地保留了重要样本的分布信息，还有效减小了计算量。

2　基于欠采样的零阶优化算法

传统的SVM算法，在对非平衡数据集进行分类时，决策超平面由于非平衡的数据分布而偏向于少数类。为此，文中引入间隔均值项的改进SVM模型，目标函数 $F(\boldsymbol{w})$ 的定义为

$$\min_{w\in R^d}F(\boldsymbol{w})=(1/2)\|\boldsymbol{w}\|^2-(\lambda_1/s)\sum_{i=1}^{s}D_i y_i\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+$$

$$(\lambda_2/s)\sum_{i=1}^{s}\max(0,1-y_i\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i) \tag{5}$$

式中：$(\lambda_1/s)\sum_{i=1}^{s}D_i y_i\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i$是引入的间隔均值项，$\lambda_1$是间隔均值项的权重参数，$\lambda_2$是损失项的权重参数，$\boldsymbol{D}$是间隔均值项的权重矩阵，是对角元素为$D_1,D_2,\cdots,D_s$的对角矩阵。对于采样集，包含多数类样本为$n_4$个，$n_2$个少数类样本。当$y_i=1$时，$D_i=n_4/n_2$；当$y_i=-1$时，$D_i=n_2/n_4$。

由代价敏感间隔分布的理论知识可知，最大化代价敏感间隔均值项可以迫使分类超平面向负类（多数类）样本偏移和旋转，有利于提高正类（少数类）样本的分类准确性。相比于式(1)标准的SVM形式，改进的算法，即式(5)更适用于非平衡数据集分类。首先，该方法引入的间隔均值项更好地考虑到了非平衡数据集的数据分布。其次，由于采样后的高非平衡比数据集并不完全是平衡的。因此，对不同类别的间隔均值项赋予了不同的权重，这样更有利于对采样后的高非平衡比数据集进行分类。

2.1 基于欠采样的线性零阶优化算法

对目标函数的求解，Pegasos算法在迭代过程中，以随机选取的单个样本梯度代替全梯度，产生了方差，影响了算法的精度和收敛速度。而文中采用了一种带有方差减小的零阶优化方法进行求解，该方法采用函数值对梯度进行估计，目标函数(5)在单个样本处的零阶梯度估计定义为

$$\hat{\nabla}F_i(\boldsymbol{w})=(\boldsymbol{u}_i/(2\mu))(F_i(\boldsymbol{w}+\mu\boldsymbol{u}_i)-F_i(\boldsymbol{w}-\mu u_i)) \tag{6}$$

式中：$\boldsymbol{u}_i$是单位超球面上依照均匀分布生成的d维向量，μ是一个接近于0的数，设定为0.1。为了减少方差的影响，使用梯度估计的修正项来减小方差：首先，在外层循环中，计算所有样本点在$\tilde{\boldsymbol{w}}_t$处的全梯度$\tilde{\boldsymbol{v}}=(1/s)\sum_{i=1}^{s}\hat{\nabla}F_i(\tilde{\boldsymbol{w}}_t)$；在内层循环中计算随机选取的单个样本在$w_k$处单个样本梯度$\hat{\nabla}F_i(\boldsymbol{w}_k)$以及在$\tilde{\boldsymbol{w}}_t$处的单个样本梯度$\hat{\nabla}F_i(\tilde{\boldsymbol{w}}_t)$。修正后的梯度估计为

$$\hat{\nabla}F_{i_k}(\boldsymbol{w}_k)-\hat{\nabla}F_{i_k}(\tilde{\boldsymbol{w}}_t)+\tilde{\boldsymbol{v}} \tag{7}$$

$\hat{\nabla}F_{i_k}(\boldsymbol{w}_k)$是在第$k$次内部循环的梯度值，后两项称为梯度修正项，基于方差减小的零阶优化算法的主要更新公式为

$$\boldsymbol{w}_{k+1}=\boldsymbol{w}_k-\eta_k(\hat{\nabla}F_{i_k}(\boldsymbol{w}_k)-\hat{\nabla}F_{i_k}(\tilde{\boldsymbol{w}}_t)+\tilde{\boldsymbol{v}}) \tag{8}$$

式中，η为步长参数，$\eta=1/(k+1)$。

结合基于几何距离加权的欠采样方法，给出基于欠采样的线性零阶优化算法(UZOVR)如下。

算法3 线性UZOVR算法

输入：采样集$\boldsymbol{S}'$，间隔均值项参数λ_1，损失函数参数λ_2，外部循环次数T，内部循环次数K。

输出：$\tilde{\boldsymbol{w}}_T$

(1) 初始化$\tilde{\boldsymbol{w}}_0$，计算权重矩阵$\boldsymbol{D}$

(2) for $t=1,2,\cdots,T$

(3) 生成方向集$\boldsymbol{u}$，并选取梯度估计方向$\boldsymbol{u}_i$

(4) $\tilde{\boldsymbol{w}}_t=\tilde{\boldsymbol{w}}_{t-1}$

(5) 计算全梯度$\tilde{\boldsymbol{v}}=(1/s)\sum_{i=1}^{s}\hat{\nabla}F_i(\tilde{\boldsymbol{w}}_t)$

(6) $\boldsymbol{w}_0=\tilde{\boldsymbol{w}}_t$

(7) for $k=1,2,\cdots,K-1$

(8) 初始化步长$\eta_k=1/(k+1)$

(9) 随机选取单个样本，通过式(6)计算单个样本点的梯度估计值

(10) 更新 $\boldsymbol{w}_{k+1}=\boldsymbol{w}_k-\eta_k(\hat{\nabla}F_{i_k}(\boldsymbol{w}_k)-\hat{\nabla}F_{i_k}(\tilde{\boldsymbol{w}}_t)+\tilde{\boldsymbol{v}})$

(11) end

(12) $\tilde{\boldsymbol{w}}_t=\boldsymbol{w}_k$

(13) end

2.2 基于欠采样的非线性零阶优化算法

对于非线性可分的问题，将原空间的样本点映射到特征空间，采用高斯核函数

$$\boldsymbol{K}(\boldsymbol{x},\boldsymbol{x}')=\langle\varphi(\boldsymbol{x}),\varphi(\boldsymbol{x}')\rangle=\exp(-\|\boldsymbol{x}-\boldsymbol{x}'\|^2/\sigma^2) \tag{9}$$

式中$\varphi(\boldsymbol{x})$是样本点$\boldsymbol{x}$通过核函数的特征映射。结合表示定理[15]，目标函数，即式(5)的最优解有如下形式

$$\boldsymbol{w}=\sum_{i=1}^{s}\boldsymbol{\alpha}_i\varphi(\boldsymbol{X}_i)=X\boldsymbol{\alpha} \tag{10}$$

式中$\boldsymbol{\alpha}^{\mathrm{T}}=[\alpha_1,\alpha_2,\cdots,\alpha_s]$，$\boldsymbol{X}=[\varphi(x_1),\varphi(x_2),\cdots,\varphi(x_s)]$。由式(9,10)，可以得到：$X^{\mathrm{T}}\boldsymbol{w}=X^{\mathrm{T}}X\boldsymbol{\alpha}=G\boldsymbol{\alpha}$，$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}=\boldsymbol{\alpha}^{\mathrm{T}}X^{\mathrm{T}}X\boldsymbol{\alpha}=\boldsymbol{\alpha}^{\mathrm{T}}G\boldsymbol{\alpha}$，$G=X^{\mathrm{T}}X$。因此，在高维特征空间空间，目标函数可表示为

$$\min_{\alpha\in R^s}F(\boldsymbol{\alpha})=\frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}G\boldsymbol{\alpha}-\frac{\lambda_1}{s}D\boldsymbol{y}^{\mathrm{T}}G\boldsymbol{\alpha}+\frac{\lambda_2}{s}\sum_{i=1}^{s}\max\{0,1-y_i\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{G}_i\} \tag{11}$$

基于欠采样的非线性零阶优化算法(UZOVR)如下。

算法4 非线性UZOVR算法。

输入：采样集$\boldsymbol{S}'$，间隔均值项参数λ_1，损失函数参数λ_2，外部循环次数T，内部循环次数K

输出：$\tilde{\boldsymbol{\alpha}}_T$

(1) 初始化$\tilde{\boldsymbol{\alpha}}_0$，计算权重矩阵$\boldsymbol{D}$

(2) for $t=1,2,\cdots,T$

(3) 生成方向集u，并选取梯度估计方向$\boldsymbol{u}_i$

(4)　$\tilde{\boldsymbol{\alpha}}_t = \tilde{\boldsymbol{\alpha}}_{t-1}$

(5)　计算全梯度 $\tilde{\boldsymbol{v}} = (1/s)\sum_{i=1}^{s}\hat{\nabla}F_i(\tilde{\boldsymbol{\alpha}}_t)$

(6)　$\boldsymbol{\alpha}_0 = \tilde{\boldsymbol{\alpha}}_t$

(7)　for　$k = 1, 2, \cdots, K-1$

(8)　　初始化步长 $\eta_k = 1/(k+1)$

(9)　　随机选取单个样本，通过式(6)计算单个样本点的梯度估计值

(10)　更　新　$\boldsymbol{\alpha}_{k+1} = \boldsymbol{\alpha}_k - \eta_k(\hat{\nabla}F_{i_k}(\boldsymbol{\alpha}_k) - \hat{\nabla}F_{i_k}(\tilde{\boldsymbol{\alpha}}_t) + \tilde{\boldsymbol{v}})$

(11)　end

(12) $\tilde{\boldsymbol{\alpha}}_t = \boldsymbol{\alpha}_k$

(13) end

算法 4 是非线性的 UZOVR 算法，主要迭代过程与算法 3 相同，迭代过程分为内外两层循环：在外部循环中计算在 $\tilde{\boldsymbol{\alpha}}_t$ 处所有样本点的全梯度，在内部循环中计算在 $\tilde{\boldsymbol{\alpha}}_t$ 处随机选取的单个样本梯度以及在 $\boldsymbol{\alpha}_k$ 处的随机选取的单个样本梯度。在两层循环中，均以最后一次迭代计算结果作为最终的输出结果。

3　实验与结果分析

本节分为 4 部分。第 1 部分对实验中所用数据集以及算法的性能评价指标进行介绍；在第 2 部分，分别给出了算法 U1ZOVR 与算法 SVM 以及算法 MWMOTE[21] 的精度对比实验；SVM 算法和 MWMOTE 算法均采用零阶随机梯度下降的方法进行求解，其主要更新公式为 $\boldsymbol{w}_{k+1} = \boldsymbol{w}_k - \eta_k\hat{\nabla}F_{i_k}(\boldsymbol{w}_k)$；U1ZOVR 则采用带有方差减小的零阶方法进行求解。在第 3 部分，给出 U2ZOVR 算法与 SVM 算法、MWMOTE 算法在精度上的对比实验；在第 4 部分是所提算法与 SVM 算法、MWMOTE 算法在运行时间上的对比实验。所有算法均采用 matlab 语言编写，版本为 matlab2017b，计算机配置为因特尔酷睿 i5-4590 处理器，3.3 GHz，8 GB 内存。

3.1　数据集与性能评价标准

在 16 个不同非平衡比数据集上测试本文提出的方法(表 1)，数据来源于 KEEL 网站[22]。其中，8 个数据集非平衡比低于 10，8 个高于 10，详细信息如下。

评价标准对于分类模型的评估至关重要。最大化精度是一般分类器的评价准则，但并不适用于非平衡数据。采用几何均值对算法精度进行评估，它的定义基于表 2 给出的混淆矩阵。

由混淆矩阵，得出几何均值(G-mean)的定义

表 1　数据集

Tab.1　Dataset

Dataset number	Dataset	Unbalanced ratio	Number	Dimension
1	Wisconsin	1.86	683	9
2	Pima	1.87	768	8
3	Vehicle1	2.90	846	18
4	Newthyroid	5.14	215	5
5	Segment0	6.02	2 308	19
6	Yeast3	8.10	1 484	8
7	Pageblocks0	8.79	5 472	10
8	Vowel0	9.98	988	13
9	Lled7digit0	10.97	443	7
10	Shuttlec0	13.87	1 829	9
11	Pageblocks1	15.86	472	10
12	Yeast4	28.1	1 484	8
13	Wine-red4	29.17	1 599	11
14	Wine-white3	44	900	11
15	Shuttle2	66.67	3 316	9
16	Poker8	85.88	1 477	10

表 2　混淆矩阵

Tab.2　Confusion matrix

Class	Predict positive class	Predict negative class
True positive class	TP	FN
True negative class	FP	TN

$$\text{G-mean} = \sqrt{\frac{\text{TP}}{\text{TP}+\text{FN}} \times \frac{\text{TN}}{\text{TN}+\text{FP}}} \tag{12}$$

为实验的公平性，所有的实验结果，采用五折交叉验证，取 5 次结果的均值作为最终结果。

3.2　算法 U1ZOVR 基于 G-mean 的对比实验

所提 U1ZOVR 算法与 SVM、MWMOTE 算法基于 G-mean 的精度对比实验。基于表 2 给出的前 8 个低非平衡比数据集，算法 U1ZOVR 与传统的 SVM 方法以及 MWMOTE 方法进行了对比实验。线性算法的训练精度和测试精度的实验结果在表 3 中给出，非线性算法的实验结果在表 4 中给出，并基于表 3 和表 4 给出了训练精度和测试精度的折线图，分别为图 1 和图 2。

由表 3，表 4 的实验结果可以看出，U1ZOVR 算法的分类精度值均高于 SVM 算法，并在大多数据集上优于 MWMOTE 算法。对比表 3，表 4 的实验结果，可以看到 3 种非线性算法的精度都高于线性算法，这是与数据集本身的特性相关，与算法无关。

通过对比图 1 中 6 种算法的折线图可以看到：对于线性分类算法，U1ZOVR 折线图的最大值为 67.44%，最低值为 53.81%；MWMOTE 算法的最

表 3 U1ZOVR 和 SVM，MWMOTE 线性算法精度对比实验

Tab.3 Accuracy comparison experiments of linear algorithm between U1ZOVR and SVM and MWMOTE

Dataset number	G-mean	Linear SVM	Linear U1ZOVR	Linear MWMOTE
1	Train	43.49	**67.44**	55.91
	Test	38.53	**64.71**	53.16
2	Train	51.06	**53.81**	50.34
	Test	50.38	**52.47**	49.07
3	Train	54.14	**58.69**	57.39
	Test	52.84	**54.29**	54.08
4	Train	56.33	59.36	**63.30**
	Test	50.44	52.80	**62.07**
5	Train	43.10	**57.72**	53.57
	Test	39.44	**57.19**	54.21
6	Train	23.24	60.13	**63.77**
	Test	18.27	57.14	**61.22**
7	Train	53.95	**61.56**	56.34
	Test	51.11	**59.41**	54.71
8	Train	52.31	**65.62**	55.32
	Test	50.21	**62.72**	54.80

表 4 U1ZOVR 和 SVM，MWMOTE 非线性算法精度对比实验

Tab.4 Accuracy comparison experiments of nonlinear algorithm between U1ZOVR and SVM and MWMOTE

Dataset number	G-mean	Kernel U1ZOVR	Kernel SVM	Kernel MWMOTE
1	Train	71.07	**82.42**	73.57
	Test	69.04	**81.55**	70.55
2	Train	63.36	**89.46**	85.18
	Test	57.9	**86.88**	83.42
3	Train	61.05	**64.34**	61.71
	Test	60.21	**62.11**	60.83
4	Train	77.85	**79.77**	73.18
	Test	71.61	**76.55**	73.05
5	Train	84.83	**86.57**	85.58
	Test	84.67	**85.54**	85.27
6	Train	62.52	**71.08**	66.64
	Test	60.78	**67.50**	67.24
7	Train	60.91	**72.61**	62.87
	Test	60.75	**72.45**	61.99
8	Train	67.89	**93.02**	65.16
	Test	65.99	**91.87**	65.80

值分别为 49.07%，63.77%；传统 SVM 算法最值分别为 56.33%，23.24%。因此，相比于波动较大的传统 SVM 算法，算法 U1ZOVR 更为稳定，但相比于算法 MWMOTE，其稳定性并未显著提高。观察非线性分类算法的折线图可以发现，3 种非线性算法波动起伏类似，能够保持较好的稳定，并且算法 U1ZOVR 相比于其他算法分类性能要高。通过图 2 亦能得到上述结论。

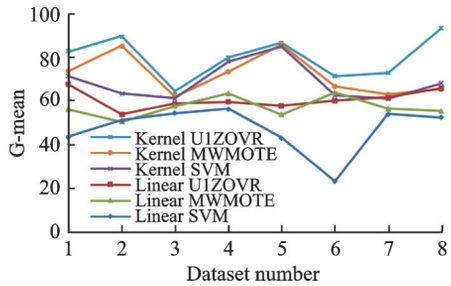

图 1 U1ZOVR 与 SVM，MWMOTE 的训练精度对比

Fig. 1 Comparison of train accuracy between U1ZOVR and SVM and MWMOTE

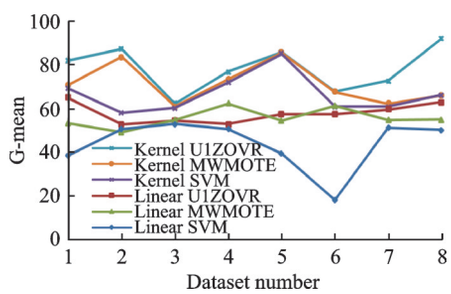

图 2 U1ZOVR 与 SVM，MWMOTE 的测试精度对比

Fig. 2 Comparison of test accuracy between U1ZOVR and SVM and MWMOTE

3.3 算法 U2ZOVR 基于 G-mean 的对比实验

基于表 2 的后 8 个高非平衡比数据集，U2ZOVR 算法与传统 SVM、MWMOTE 进行了对比实验。线性算法的测试精度和训练精度的实验结果在表 5 中给出，非线性算法的实验结果为表 6，图 3 和图 4 分别为基于表 5 和表 6 的训练精度和测试精度的折线图。

表 5，表 6 给出了 6 种算法基于 G-mean 的实验结果，算法 U2ZOVR 的分类性能明显优于 SVM 算法、MWMOTE 算法，这充分说明了算法 U2ZOVR 在高非平衡比数据集上的优势。由表 3，表 4 的实验结果可以看到，在大多数数据集上 3 种非线性算法的实验结果明显优于线性算法；而在 Page-blocks1 数据集上，非线性算法的实验结果低于线性算法的。

对比图 3 中的 6 种算法，可以看到 SVM 算法在数据集 Pageblocks1，Shuttlec0 以及 Wine-white3 上存在较大的起伏；MWMOTE，U2ZOVR 算法仅在数据集 Pageblocks1 上波动较大，而在其他数据集上算法 U2ZOVR 的波动幅度比算法 MWMOTE 要小。通过以上分析，在高非平衡比数据

表 5　U2ZOVR 和 SVM，MWMOTE 线性算法精度对比实验

Tab.5　Accuracy comparison experiments of f linear algorithm between U2ZOVR and SVM and MWMOTE

Dataset number	G-mean	Linear SVM	Linear U2ZOVR	Linear MWMOTE
9	Train	54.89	63.37	**64.12**
	Test	58.62	60.93	**60.98**
10	Train	70.89	**72.02**	68.11
	Test	60.55	**61.30**	58.32
11	Train	53.37	**81.39**	65.30
	Test	56.96	**79.27**	64.65
12	Train	16.44	60.45	**65.72**
	Test	8.51	53.35	**59.86**
13	Train	41.85	**57.89**	55.53
	Test	36.89	**56.91**	54.04
14	Train	51.29	**56.25**	55.74
	Test	42.49	**53.93**	53.16
15	Train	44.21	**67.17**	59.96
	Test	35.90	**61.83**	56.09
16	Train	41.37	**53.77**	52.40
	Test	40.72	**52.86**	51.40

表 6　U2ZOVR 和 SVM,MWMOTE 非线性算法精度对比实验

Tab.6　Accuracy comparison experiments of nonlinear algorithm between U2ZOVR and SVM and MWMOTE

Dataset number	G-mean	Kernel SVM	Kernel U2ZOVR	Kernel MWMOTE
9	Train	65.06	**75.35**	66.35
	Test	61.60	**74.54**	64.86
10	Train	73.58	**81.47**	76.64
	Test	67.28	**79.78**	75.14
11	Train	35.60	**52.92**	39.60
	Test	33.96	**50.03**	40.59
12	Train	62.78	**83.87**	75.60
	Test	59.30	**82.96**	74.94
13	Train	61.96	**79.64**	74.59
	Test	60.16	**78.98**	74.85
14	Train	78.92	**83.85**	83.58
	Test	76.17	**83.31**	76.23
15	Train	73.82	**86.94**	80.78
	Test	71.67	**85.96**	79.29
16	Train	72.99	**88.28**	83.76
	Test	70.23	**86.35**	84.14

集上，相比于传统的 SVM 算法，U2ZOVR 算法能够表现出较好的分类性能及稳定性；与 MWMOTE 算法相比，算法 U2ZOVR 的分类精度优于

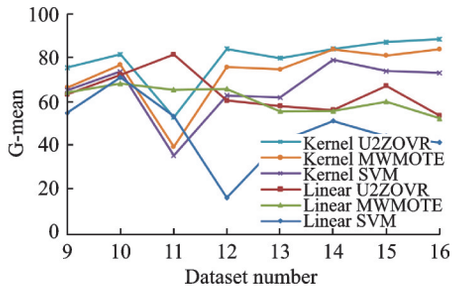

图 3　U2ZOVR 与 SVM，MWMOTE 的训练精度对比

Fig. 3　Comparison of train accuracy between U2ZOVR and SVM and MWMOTE

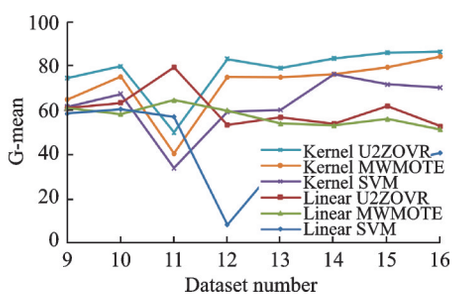

图 4　U2ZOVR 与 SVM，MWSMOTE 的测试精度对比

Fig. 4　Comparison of test accuracy between U2ZOVR and SVM and MWMOTE

SMOTE，但是算法 MWMOTE 的稳定性略优于算法 U2ZOVR，稳定性没有得到较大提升。

3.4　时间性能对比实验

在本节中，给出 U1ZOVR 算法、U2ZOVR 算法分别与传统 SVM 算法、MWMOTE 算法在总运行时间的对比实验。总运行时间包括训练时间以及测试时间，实验结果均采用五折交叉验证的均值作为最终结果，时间单位为秒(s)。算法 U1ZOVR 实验结果在表 7，表 8 中给出，算法 U2ZOVR 实验结果在表 9，表 10 中给出。

在上述实验结果中，基于欠采样的零阶优化算法在所有的实验数据集上，其总运行时间明显比传

表 7　U1ZOVR 和 SVM，MWMOTE 线性算法时间对比实验

Tab.7　Time comparison experiments of linear algorithm between U1ZOVR and SVM and MWMOTE　s

Dataset number	Linear SVM	Linear U1ZOVR	Linear MWMOTE
1	0.45	**0.27**	1.20
2	0.43	**0.23**	1.14
3	0.78	**0.26**	1.85
4	0.21	**0.20**	0.48
5	1.73	**0.33**	5.24
6	0.68	**0.27**	1.92
7	2.01	**0.43**	19.32
8	0.70	**0.24**	1.34

表8 U1ZOVR和SVM,MWMOTE非线性算法时间对比实验

Tab.8 Time comparison experiments of nonlinear algorithm between U1ZOVR and SVM and MWMOTE s

Dataset number	Kernel SVM	Kernel U1ZOVR	Kernel MWMOTE
1	30.86	**8.92**	40.16
2	41.09	**19.41**	108.51
3	50.79	**16.18**	115.71
4	1.69	**0.58**	2.19
5	402.35	**108.79**	722.78
6	167.02	**29.15**	567.87
7	2 327.38	**574.82**	5 790.80
8	70.19	**12.90**	190.53

表9 U2ZOVR和SVM,MWMOTE线性算法时间对比实验

Tab.9 Time comparison experiments of linear algorithm between U2ZOVR and SVM and MWMOTE s

Dataset number	Linear SVM	Linear U2ZOVR	Linear MWMOTE
9	0.28	**0.23**	0.62
10	0.86	**0.28**	1.51
11	0.40	**0.24**	0.71
12	0.71	**0.25**	1.31
13	0.91	**0.26**	1.62
14	0.58	**0.23**	1.07
15	1.34	**0.25**	1.89
16	0.80	**0.24**	1.44

表10 U2ZOVR和SVM,MWMOTE非线性时间对比实验

Tab.10 Time comparison experiments of nonlinear algorithm between U2ZOVR and SVM and MWMOTE s

Dataset number	Kernel SVM	Kernel U2ZOVR	Kernel MWMOTE
9	5.48	**0.76**	24.39
10	262.42	**23.99**	392.57
11	14.24	**0.87**	25.17
12	170.89	**1.14**	367.75
13	198.39	**1.24**	436.26
14	60.10	**0.72**	140.15
15	848.08	**1.31**	1 039.33
16	170.76	**0.78**	4 431.43

统的SVM算法以及MWMOTE算法要少。并且,对于非线性算法,这种优势更加明显。由此,可以充分说明基于欠采样的零阶优化算法可以减少大量的计算,有效节省了运行时间。

对比U1ZOVR与U2ZOVR的运行时间,可以发现算法U2ZOVR运行时间较短相比于算法U1ZOVR,在非线性算法上更为突出。例如,在样本个数相同的数据集Yeast3,Yeast4上,两种非线性算法的运行时间为29.15,1.14。这是由于算法U2ZOVR仅对距离超平面较近的多数类样本(数量为少数类样本10倍的多数类样本)进行采样,而算法U1ZOVR对则对所有的多数类样本进行采样,采样后样本数量差别较大,导致了两种算法运行时间的差异。

在表10中,显然算法U2ZOVR在数据集Shuttlec0上的运行时间明显要比在其它数据集上要长,这亦是由于算法U2ZOVR仅对距离超平面较近的多数类样本(数量为少数类样本10倍的多数类样本)进行采样的原因。以数据集Shuttlec0,Shuttle2为例,它们的总样本个数分别为1 829,3 316,少数类样本个数为123,49。对10倍的少数类样本进行采样后,显然Shuttlec0的采样集样本的个数要多于Shuttle2,因此其运行时间要长。

最后,对算法U1和算法U2的时间复杂度进行分析:由算法1的伪代码进行计算,可以得出算法1的时间复杂度为$n_1\times d+n_1+C$。其中,d为数据集的维数,n_1是多数类样本个数,C为常数;由算法2的伪代码进行计算,可知算法2的时间复杂度为$n_1\times d+10\times n_2+C$。其中,$n_1$是多数类样本个数,$n_2$是少数类样本个数,$C$为常数。由上述分析,可以得出两种算法的时间复杂度均与数据集的多数类样本个数、少数类样本个数以及维数紧密相关。

由于,本文在算法2的实验中设定数量为少数类样本10倍的多数类样本。所以,在相同的非平衡比高于10的数据集上,由于$n_1>10\times n_2$,算法1的时间复杂度高于算法2;但是,在相同的非平衡比低于10的数据集上,由于10倍的n_2大于多数类的样本个数,已经违背了数据的非平衡信息,算法2的实验设定不再适用,这时需要重新设定需保留的少数类数类的个数。

4 结 论

针对不同非平衡比的数据集,提出了基于欠采样的零阶优化算法。通过使用加权欠采样对数据进行预处理,以及在优化模型中引入间隔均值项,削弱了非平衡分布对分类的影响;使用带有方差减小的求解方法,减小了随机迭代过程产生的方差。在16个不同非平衡比数据集上的实验结果表明:该方法在大多情况下,有较好的分类结果,并且可以有效地降低计算量,提高收敛速度。

参考文献:

[1] OFEK N, ROKACH L, STERN R, et al. Fast-CBUS: A fast clustering-based undersampling method for addressing the class imbalanced problem[J]. Neurocomputing, 2017, 243: 88-102.

[2] MATHEW J, PANG C K, LUO Ming, et al. Classification of imbalanced data by oversampling in kernel space of support vector machines[J]. IEEE Trans Neural Netw Learn Systems, 2018, 29(9): 4065-4076.

[3] WONG G Y, LEUNG F H F, LING S H. A hybrid evolutionary preprocessing method for imbalanced datasets[J]. Information Sciences, 2018, 454/455: 161-177.

[4] ZHANG Chong, TAN K C, LI Haizhou, et al. A cost-sensitive deep belief network for imbalanced classification[J]. IEEE Transactions on Neural Networks and Learning Systems, 2019, 30(1): 109-122.

[5] TUNTIWACHIRATRAKUN P, VATEEKUL P. Applying active learning strategy to classify large scale data with imbalanced classes[C]//International Conference on Control. South Korea, Mr Andreessen: IEEE Computer Society Press, 2017: 100-105.

[6] ERFANI S M, REJASEGARAR S, KARUNASEKERA S, et al. High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning[J]. Pattern Recognition, 2016, 58(C): 121-134.

[7] LU Wei, LI Zhe, CHU Jinghui. Adaptive ensemble under sampling-boost: A novel learning framework for imbalanced data[J].Journal of Systems and Software, 2017,132: 272-282.

[8] BAO Lei, CAO Juan, LI Jintao.Boosted near-miss under-sampling on SVM ensembles for concept detection in large-scale imbalanced data sets[J]. Neurocomputing, 2016, 172: 198-206.

[9] COLLELL G, PRELEC D, PATIL K R.A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data [J].Neurocomputing, 2018, 275: 330-340.

[10] GHAZIKHANI A, YAZDI H S, MONSEFI R. Class imbalance handling using wrapper-based random oversampling[C]//Electrical Engineering (ICEE), 20th Iranian Conference on IEEE. Tehran, Iran: IEEE, 2012: 611-616.

[11] ZHU Tuanfei, LIN Yaping, LIU Yonghe. Synthetic minority oversampling technique for multiclass imbalance problems[J]. Pattern Recognition, 2017, 72: 327-340.

[12] LIN Weichao, TSAI C F, HU Y H, et al. Clustering-based under-sampling in class-imbalanced data[J]. Information Sciences, 2017, 409: 17-26.

[13] QI Kang, LEI Shi, ZHOU Mengchu, et al. A distance-based weighted undersampling scheme for support vector machines and its application to imbalanced classi fi cation[J]. IEEE Transactions on Neural Networks and Learning Systems, 2017, 99: 1-14.

[14] CORTES C, VAPNIK V. Support vector networks[J]. Machine Learning, 1995, 20(3): 273-297.

[15] CHENG Fanyong, ZHANG Jing, WEN Cuihong, et al. Large cost-sensitive margin distribution machine for imbalanced data classification[J]. Neurocomputing, 2017, 224: 45-57.

[16] SHALEV-SHWARTZ S, SINGER Y, SREBRO N. Pegasos: Primal estimated sub-gradient solver for SVM[J]. Mathematical Programming, 2011, 127(1): 3-30.

[17] GHADIMI S, LAN G. Stochastic first and zeroth-order methods for nonconvex stochastic programming[J]. Siam Journal on Optimization, 2013, 23(4): 2341-2368.

[18] LI Jueyou, WU Changzhi, WU Zhiyou, et al. Gradient-free method for nonsmooth distributed optimization[J]. Journal of Global Optimization, 2015, 61(2): 325-340.

[19] JOHNSON R, ZHANG Tong. Accelerating stochastic gradient descent using predictive variance reduction[C]//Advanced in Neural Information Processing Systems. California, America: Curran Associates Inc, 2013: 315-323.

[20] LIAN Xiangru, ZHANG Huan, HSIEH C J, et al. A comprehensive linear speedup analysis for asynchronous stochastic parallel optimization from zeroth-order to first-order[C]//30th Conference on Neural Information Processing Systems(NIPS 2016). Barcelona,Spain: Curran Associates Inc, 2016: 1-9.

[21] BARU S, ISLAM M M, YAO X, et al. MWMOTE—Majority weighted minority oversampling technique for imbalanced data set learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(2): 405-425.

[22] VAIRAGADE M. Keel: A software tool to assess evolutionay algorithmms for Data Mining problems [EB/OL]. (2003-10-17)[2019-02-20]. http://www.keel.es/.

(编辑:刘彦东)