

引用格式:柳毅,曾昊. 改进 K-means 的双向采样非均衡数据分类方法[J]. 微电子学与计算机,2020,37(3):60-65. [LIU Y, ZENG H. Improved the bi-directional sampling unbalanced data classification method of k-means [J]. Microelectronics & Computer,2020,37(3):60-65.]

改进 K-means 的双向采样非均衡数据分类方法

柳毅,曾昊

(广东工业大学 计算机学院,广东 广州 510006)

摘 要: 针对分类器在不均衡数据集上对小类分类准确率较差的问题,提出了改进 K-means 的双向采样算法 KMBS(k-means bi-directional sampling),并将集成学习应用在分类算法上. 首先,使用改进的 K-means 聚类算法将原始数据集划分为不同的聚类簇. 其次,在聚类簇中使用改进的 SMOTE 算法对小类样本过采样,对聚类簇内的大类样本欠采样,使数据集平衡. 多次执行该算法可以产生多个差异较大的数据集,因此训练出多个差异较大的分类器,提升集成学习的效果. 通过分析实验结果,该算法较现有几种算法不仅能提高整体分类性能,并且有效提高小类样本的分类性能.

关键词: 不均衡学习;双向采样;分类算法;集成学习

中图分类号: TP391

文献标识码: A

文章编号: 1000—7180(2020)03—0060—06

Improved the bi-directional sampling unbalanced data classification method of K-means

LIU Yi,ZENG Hao

(College of Computer Science, Guangdong University of Technology, Guangzhou 510006, China)

Abstract: Aiming at the poor classification accuracy of minority classes by classifier on unbalanced data sets, an improved k-means bi-directional sampling algorithm KMBS (k-means bi-directional sampling) is proposed, and integrated learning is applied to the classification algorithm. First, the improved k-means clustering algorithm is used to divide the original data set into different clustering clusters. Secondly, oversampling of the minority and under-sampling of the majority in the cluster using the modified SMOTE algorithm in the cluster, so as to make the dataset balance. Multiple executions of this algorithm can produce multiple data sets with large differences, so multiple classifiers with large differences can be trained to improve the effect of ensemble learning. By analyzing the experimental results, this algorithm can not only improve the overall classification performance, but also improve the classification performance of a few kinds of samples.

Key words: imbalanced learning; bi-directional sampling; classification; ensemble learning

1 引言

机器学习中的数据类别不均衡问题描述的是在分类问题中不同类别样本数量的比例相差过大的问题^[1]. 在实际应用中^[2-6],由于分类算法的优化目标

函数是准确率,会造成大类吞噬小类的现象. 比如在 100 个数据点中,大类样本数量为 90 个,小类样本数量为 10 个,即使将所有小类样本预测为大类,准确率也能达到 90%.

大类吞噬小类的根本原因在于小类样本数量的

稀少性。因此,许多研究者在 SMOTE 算法的基础上,以提升小类样本数量为目标,提出了许多方法^[7-11]。但这些方法会容易生成噪声样本点,冗余样本点,没有从过采样和欠采样两个方面考虑。

本文提出了一种基于改进 K-means 的双向采样算法(KMBS)。双向采样即为,在增加小类样本的数量下(过采样),随机适当的减少大类样本的数量(欠采样),使得大类样本和小类样本数量达到平衡。由于在减少大类样本的情况下会失去一部分样本信息。因此 KMBS 算法引入了集成学习的机制。通过多次合成小类样本和随机选择大类样本组成多个差异较大的训练集,训练多个分类器。可以减少大类样本信息的丢失,并且提升分类性能。

2 改进 K-means 算法

K-means 算法是最简单经典的聚类算法,有着广泛的应用。K-means 算法初始阶段需要根据超参数 K 随机选取 K 个初始的聚类中心。但正是由于初始聚类中心选择的随机性,使得最终聚类结果不稳定,准确性不高。理想的聚类中心点是自身周围样本密度较高,而各个聚类中心点的距离较大的点。为了寻找出这样的中心点,引入如下定义和概念:

任意两点 $\vec{x}_i, \vec{x}_j$ 的欧式距离:设样本点为 n 维向量 $\vec{x} = (x^1, x^2, \dots, x^n)$, 则任意两个样本点 $\vec{x}_i, \vec{x}_j$ 之间的欧式距离为

$$\text{dist}(\vec{x}_i, \vec{x}_j) = \sqrt{\sum_{k=1}^n (\vec{x}_i^k - \vec{x}_j^k)^2} \quad (1)$$

全局平均半径 $\bar{R}$: 设数据集共有 M 个样本点, 则

$$\bar{R} = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^m d(\vec{x}_i, \vec{x}_j) \quad (2)$$

任一数据点 $\vec{x}_i$ 的点数密度 $\Omega(\vec{x}_i)$: 假设任意一个数据样本点为 $\vec{x}_i$, 以 $\vec{x}_i$ 为中心, 半径长度为全局平均半径 $\bar{R}$ 的圆域内, 其包含样本点的个数即为 $\vec{x}_i$ 的点数密度

$$\Omega(\vec{x}_i) = \sum_{j=1}^m f(\text{dist}(\vec{x}_i, \vec{x}_j) < \bar{R}) \quad (3)$$

任意一数据点 $\vec{x}_i$ 的领域半径 $R(\vec{x}_i)$: 假设在以数据点 $\vec{x}_i$ 为中心, 半径为 $\bar{R}$ 的圆域内有 k 个样本点 $\vec{x}_1, \dots, \vec{x}_k$, 则

$$R(\vec{x}_i) = \frac{1}{k} \sum_{j=1}^k \text{dist}(\vec{x}_i, \vec{x}_j) \quad (4)$$

任一点 $\vec{x}_i$ 的领域密度 $\rho(\vec{x}_i)$:

$$\rho(\vec{x}_i) = \frac{\Omega(\vec{x}_i)}{R(\vec{x}_i)} \quad (5)$$

任一点 $\vec{x}_i$ 的距离散度 $\psi(\vec{x}_i)$: 对于点 $\vec{x}_i$ 假设其点数密度为 $\Omega(\vec{x}_i)$, 在以 $\vec{x}_i$ 为中心, $\bar{R}$ 为半径的圆域内, 若存在一点 $\vec{x}_j$, 其点数密度 $\Omega(\vec{x}_j) > \Omega(\vec{x}_i)$, 则

$$\psi(\vec{x}_i) = \min(\text{dist}(\vec{x}_i, \vec{x}_j)) \quad (6)$$

若不存在这样的点, 则

$$\psi(\vec{x}_i) = \max(\text{dist}(\vec{x}_i, \vec{x}_j)) \quad (7)$$

任一数据点 $\vec{x}_i$ 的中心指数 $\xi(\vec{x}_i)$:

$$\xi(\vec{x}_i) = \rho(\vec{x}_i) \times \psi(\vec{x}_i) \quad (8)$$

在上述定义中 $\bar{R}$ 代表了整个数据集中点与点之间的平均距离, $\Omega(\vec{x}_i)$ 代表了在这种平均距离下, 点 $\vec{x}_i$ 的周围点的个数, $R(\vec{x}_i)$ 代表点 $\vec{x}_i$ 与周围点的紧密程度, $\rho(\vec{x}_i)$ 等于 $\Omega(\vec{x}_i)$ 与 $R(\vec{x}_i)$ 的商, $\rho(\vec{x}_i)$ 越大表示 $\vec{x}_i$ 的自身密度越大, 而距离散度 $\psi(\vec{x}_i)$ 则度量了其与其他待定聚类中心的距离程度, 中心指数 $\xi(\vec{x}_i)$ 为领域密度和距离散度之积, 则中心指数越大, 代表点 $\vec{x}_i$ 自身周围样本密度越高, 与其他高密度样本点距离越大, 可以很好的度量一个样本点是否适合做初始聚类中心。

3 KMBS 算法

3.1 改进的 K-means 算法

KMBS 算法首先在数据集上使用改进的 K-means 算法将数据集分成多个不同的聚类簇。改进的 K-means 算法与传统的 K-means 算法区别在于不再是根据聚类个数 K 随机的选择 K 个初始聚类中心, 而是根据公式 8 计算每个数据点的中心指数, 选择中心指数最大的 K 个样本点作为初始化的聚类中心。

经过改进的 K-means 算法聚类后, 数据集被划分成三种不同的聚类簇:

(1) 只包含大类样本点的聚类簇, 无需过采样, 待欠采样。

(2) 只包含小类样本点的聚类簇, 待过采样。

(3) 同时包含大类样本点和小类样本点的聚类簇, 则该聚类簇可能包含了决策边界, 该聚类簇待过采样和欠采样。

3.2 过采样

经过聚类之后, 进入 SMOTE^[11] 过采样阶段。传统的 SMOTE 算法对于某个待采样样本点 $X = (x_1, x_2, \dots, x_n)$, 计算 X 与其他样本点 $X' = (x'_1,$

$x'_2, \dots, x'_n$) 的欧氏距离:

$$\text{dist}(X, X') = \sqrt{\sum_{i=1}^n (x_i - x'_i)^2} \quad (9)$$

欧式距离最小的 K 个样本点为 X 的 K 个近邻样本点, 设过采样率为 N (过采样成合成样本个数), 随机从 K 个近邻样本点选择一个样本点 $Y = (y_1, y_2, \dots, y_n)$, 按如下公式合成样本点:

$$X_{\text{syn}} = X + \text{rand}(0, 1) * (Y - X) \quad (10)$$

其中 $\text{rand}(0, 1)$ 代表在闭区间 $(0, 1)$ 内随机生成的小数, 按以上方法重复执行 N 次, 直到达到过采样率 N . 该方法简单方便、但同时也存在缺陷, 合成的样本点线性相关, 如图 1 所示.

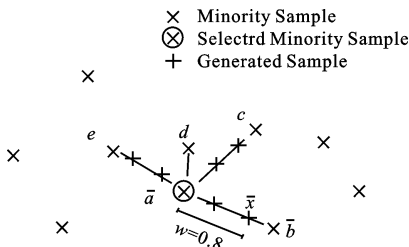


图 1 传统 SMOTE 算法缺陷示意图

当过采样率 N 大于近邻样本点个数 K 时, 待采样点 a 某个近邻样本点, 比如 b 被重复选择, 根据公式 10, a 和 b 合成的样本点只会出现在 a 和 b 之间的直线上. 然而在过采样时, 理想的情况是生成的样本点能够随机均匀的分布在采样区域里.

针对上述问题, KMBS 算法在过采样阶段改进了传统的 SMOTE 算法. 首先引入边界小类样本点和内部小类样本点两个定义:

(1) 内部小类样本点: 在某聚类簇中, 某个小类样本点的 $2k+1$ 个近邻样本点中, 小类样本点个数 $m, k < m \leq 2k+1$, 则该样本点为内部小类样本点.

(2) 边界小类样本点: 在某个聚类簇中, 某个小类样本点的 $2k+1$ 个近邻样本点中, 小类样本点个数 $m, 0 < m \leq k$, 则该样本点为边界小类样本点.

对于内部小类样本点 X :

① 按照公式寻找 k 个近邻样本点

② 随机选择两个不同的近邻点 Y_1, Y_2

③ 按照下式生成中间过渡样本点:

$$Y' = Y_1 + \text{rand}(0, 1) * (Y_2 - Y_1) \quad (11)$$

④ 按下式合成样本点:

$$X_{\text{synth}} = X + \text{rand}(0, 1) * (Y' - X) \quad (12)$$

对于边界小类样本点 X :

① 按照公式寻找其 K 个近邻样本点;

② 随机选择两个不同的近邻样本点 Y_1, Y_2

③ 按照公式 (11) 生成中间过渡样本点 Y' 若 Y_1, Y_2 都是小类样本点则按公式 (12) 合成样本点, 否则按下式合成:

$$X_{\text{synth}} = X + \text{rand}(0, 0.5) * (Y' - X) \quad (13)$$

如图 2 所示, 合成的样本均匀分布在区域内.

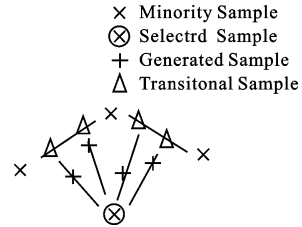


图 2 改进的 SMOTE 算法效果图

公式 (12) 和公式 (13) 的不同之处在于随机数的取值范围不同, 当 Y_1, Y_2 为大类样本点时, 为使合成样本点更加靠近小类边界, 将随机数取值设置为 $0 \sim 0.5$ 之间.

3.3 欠采样

当所有的待过采样聚类簇完成过采样后, 开始欠采样. 欠采样的意义不仅在于减少大类样本的数量可以减少模型训练时间, 当大类样本和小类样本比例严重失调时, 过高的过采样率使大类和小类数量平衡会使训练的模型存在过拟合问题.

最常用的欠采样方法是随机欠采样, 随机的从样本中抽取一定数量的样本. 该方法的全局随机性在采样率较小时, 往往偏向于数据样本集中、密度较大的区域, 忽略较稀疏的区域, 而往往这些较为稀疏的区域可能是边界区域.

KMBS 算法将数据集分成若干个聚类簇, 在聚类簇中进行局部的欠采样可以保证不同区域都可以被采样到部分样本点, 避免了全局随机欠采样带来的缺陷.

(1) 欠采样: 对每一个待欠采样聚类簇 cluster, 设其聚类簇内样本个数为 $|\text{cluster}|$, 欠采样率为 N , 则该聚类簇经欠采样后包含的大类样本点个数为 $\text{ceil}(|\text{cluster}| * N)$.

(2) 在 cluster 中, 随机不重复的选择一个大类样本点加入采样集合, 直到数量达到 $\text{ceil}(|\text{cluster}| * N)$, 欠采样完成.

3.4 集成学习

集成学习的主要思路是先通过一定的规则生成多个学习器, 再采用某种集成策略进行组合, 最后综合判断输出最终结果. 一般而言, 通常所说的集成学习中的多个学习器都是同质的 "弱学习器". 该弱学

习器,通过样本集扰动、输入特征扰动、输出表示扰动、算法参数扰动等方式生成多个学习器,进行集成后获得一个精度较好的强学习器。

KMBS 算法在欠采样中近邻样本点的随机选择,样本合成公式中的随机数,以及欠采样阶段的随机选择,使得每次执行 KMBS 算法得到的数据集差异性较大,训练得到的分类器具有差异性,采用集成学习能得到提升.集成的策略采用投票表决的方式:

$$C(x) = \operatorname{argmax}_c \sum_i \operatorname{classifier}_i(x)$$

式中: x 代表待分类样本点, $\operatorname{classifier}_i(x)$ 代表第 i 个弱分类器的预测值, $C(x)$ 代表集成学习最终输出结果。

4 实验结果分析

为了证明 KMBS 算法的有效性,在实际的不平衡数据集上,与 SMOTE 算法^[11],BSMOTE 算法^[12],KM-SMOTE 算法^[13],以及在不对不平衡数据集进行任何采样处理的情况下,选取五个常用的分类器,支持向量机(SVM),K 近邻分类器(KNN),逻辑斯蒂回归分类器(LOGIST),决策树分类器(DT),朴素贝叶斯分类器(NB)进行实验对比分析,得出本文算法比上述几种算法更优^[14-15]。

4.1 数据集

本文从 UCI 机器学习数据选取三个常用的数据集进行测试,分别为 abalone, glass, yeast. 数据集的基本信息如表 1 所示。

表 1 数据集的基本信息

数据集	属性数量	小类样本数量	大类样本数量
Abalone	8	490	3 687
Yeast	8	187	1 297
Glass	10	51	163

4.2 实验评价指标

常用的分类器性能度量指标比如准确率,在数据集不平衡的情况下是不适用的,但是混淆矩阵,如表 2 所示,具有评价一个分类器在不均衡数据集上性能的所有信息。

表 2 混淆矩阵

类别	预测正样本	预测负样本
真正正样本	TP	FN
真实负样本	FP	TN

列值代表预测类标签,行值代表真实类标签,大类样本的标签为负(negative),小类的标签为正

(positive). TP 代表所有预测为正样本的数据中,其真实标签为正的数量;FP 表示所有预测为正样本的标签,中其真实标签为负的数量;TN 代表所有预测为负样本的标签中,其中真实标签为负的样本数量;FN 代表所有预测为负样本的标签中,其真实标签为正的样本数量。

研究人员基于混淆矩阵提出许多评估准则,G-mean 和 F-measure 是度量分类器在不均衡数据集上性能最常用的两种方法。

G-means 是一种衡量分类器在数据集整体上分类性能的标准,综合考虑了分类器对大类和小类的分类性能,计算公式如下:

$$\text{G-means} = \sqrt{\text{TPR} \times \text{FPR}} \tag{14}$$

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{15}$$

$$\text{FPR} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{16}$$

当分类器偏向于某一类时,G-means 的值将很小. F-measure 是针对分类器对小类样本分类性能的指标,其计算公式如下:

$$\text{F-measure} = \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{17}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{18}$$

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{19}$$

一个完美无分类错误的分类器其 $\text{recall} = 1$, $\text{precision} = 1$,此时 F-measure 值最大为 0.5。

4.3 参数设置

KMBS 算法需设置三个参数,改进的 K-means 算法聚类簇个数 C, SMOTE 近邻样本点个数 K,以及欠采样倍率 R. 实验中采用控制变量法,即控制其中两个变量值不变,另一个变量值改变,观察 KMBS 算法 G-means 和 F-measure 的变化,使 G-means 和 F-measure 值达到最大的变量值为最佳变量值. 经实验,五个分类器在三个数据集上的最佳的 C 值, K 值, 以及 R 值如表 3~表 6 所示。

表 3 五个分类器在三个数据集上最优 C 值

	Yeast	Abalone	Glass
KNN	8	11	7
Logist	10	8	10
DT	10	11	6
SVM	3	8	7
NB	2	5	7

表 4 五个分类器在三个数据集上最优 K 值

	Yeast	Abalone	Glass
KNN	7	9	5
Logist	7	8	5
DT	6	10	6
SVM	7	10	7
NB	6	6	5

表 5 五个分类器在三个数据集上最优 R 值

	Yeast	Abalone	Glass
KNN	0.6	0.5	0.5
Logist	0.6	0.5	0.4
DT	0.5	0.4	0.7
SVM	0.6	0.5	0.4
NB	0.6	0.4	0.4

4.4 不同算法的性能比较

本文实验环境为 windows10,python3.6。由于 SMOTE 算法^[11],BSMOTE 算法^[12],KMSMOTE 算法^[13],KMBS 算法是对不均衡数据集在数据层面的处理,与具体分类算法无关,为充分验证算法的有效性,从 scikit-learn 机器学习库中选取五个常用的分类算法,K 近邻分类算法(KNN),逻辑斯蒂回归分类算法(Logist),决策树分类算法(DT),支持向量机(SVM),贝叶斯分类算法(NB),分类算法的参数均为标准默认参数;每次实验随机选取 75% 的样本作为训练数据,剩余 25% 的样本作为测试数据;评价指标采用 F-measure 和 G-means;集成学习每次训练相同算法的集分类器,避免因分类器不同带来的扰动,集成学习的基分类器个数设置为 10 个;试验次数为 500 次,取 500 次实验的平均值为最终实验结果,如表 6~8 所示。

表 6 yeast

算法	Base		SMOTE		BSMOTE		KMSMOTE		KMBS	
	F-measure	G-means	F-measure	G-means	F-measure	G-means	F-measure	G-means	F-measure	G-means
KNN	0.267 4	0.725 4	0.323 1	0.812 6	0.364 2	0.821 2	0.357 3	0.816 5	0.378 3	0.842 4
Logist	0.275 7	0.741 3	0.287 1	0.767 4	0.356 3	0.840 7	0.344 3	0.812 9	0.334 3	0.839 9
DT	0.234 9	0.713 6	0.306 5	0.774 3	0.304 6	0.785 4	0.317 0	0.794 1	0.336 4	0.831 8
SVM	0.131 2	0.383 6	0.302 4	0.787 1	0.362 2	0.844 0	0.328 8	0.817 4	0.387 4	0.879 6
NB	0.309 1	0.729 2	0.328 9	0.800 1	0.323 0	0.812 9	0.345 8	0.820 5	0.361 7	0.857 7

表 7 abalone

算法	Base		SMOTE		BSMOTE		KMSMOTE		KMBS	
	F-measure	G-means	F-measure	G-means	F-measure	G-means	F-measure	G-means	F-measure	G-means
KNN	0.221 1	0.655 6	0.297 0	0.712 3	0.314 5	0.753 1	0.326 4	0.774 6	0.316 1	0.745 7
Logist	0.217 1	0.634 7	0.266 7	0.666 8	0.260 4	0.654 3	0.257 6	0.644 0	0.287 2	0.691 4
DT	0.231 9	0.687 7	0.284 9	0.723 8	0.329 6	0.763 1	0.337 3	0.786 2	0.352 4	0.821 4
SVM	0.115 6	0.345 5	0.298 4	0.691 8	0.318 3	0.734 7	0.316 6	0.743 9	0.336 8	0.751 4
NB	0.180 8	0.628 6	0.275 8	0.710 7	0.283 4	0.725 4	0.297 1	0.740 4	0.284 9	0.749 2

表 8 glass

算法	Base		SMOTE		BSMOTE		KMSMOTE		KMBS	
	F-measure	G-means	F-measure	G-means	F-measure	G-means	F-measure	G-means	F-measure	G-means
KNN	0.407 1	0.864 7	0.416 5	0.893 4	0.415 4	0.916 7	0.417 6	0.914 5	0.437 6	0.934 8
Logist	0.388 9	0.819 9	0.398 1	0.851 0	0.401 3	0.855 0	0.408 9	0.889 6	0.425 4	0.908 6
DT	0.402 3	0.851 9	0.406 7	0.883 8	0.413 6	0.898 2	0.419 2	0.898 5	0.439 2	0.919 3
SVM	0.377 6	0.826 7	0.391 2	0.833 4	0.399 5	0.844 3	0.404 7	0.877 3	0.413 4	0.897 6
NB	0.397 2	0.840 8	0.410 9	0.874 0	0.411 3	0.884 4	0.417 5	0.865 4	0.437 7	0.936 9

KMBS 算法在表 6 中有四次 F-measure 和 G-means 值最高,在表 7 中有 3 次 F-measure 和 G-

means 值最高,且性能提升较为明显,在不是最高的两次情况下,也接近最高值. 因为 KMBS 算法综合考虑了边界样本的重要性,以及避免产生噪声、入侵点. 在表 8 中, KMBS 算法有五次 F-measure 和 G-means 值最高,性能有较明显的提高这是因为在数据量小的情况下,集成学习比单个分类器能学习到更多的数据信息,提升性能.

5 结束语

为了解决不均衡数据集带来的分类器偏向大类问题,本文提出了 KMBS 算法. 该算法改进了 K-Means 算法,使聚类更加精准;在聚类簇中对待采样的小类样本点分为内部小类样本和边界小类样本,考虑了边界样本点在决策中的重要性;在合成少数类样本阶段,采用了新的合成方法,避免了原有方法带来的线性问题,使合成样本分布更为理想;欠采样处理使得算法产生的数据集更具有差异性,提升集成学习的效果. 通过选取五个常用的机器学习分类算法,在三个 UCI 数据集上的对比实验表明, KMBS 算法具有更好的分类性能. 然而,本文算法在集成学习中只使用了同种分类算法的基分类器,并没有考虑不同算法的分类器集成,因此下一步本文将进一步研究不同分类器集成的效果,以期进一步提升算法性能.

参考文献:

- [1] WEISS G M, ZADROZNY B, SAAR M. Guest editorial: special issue on utility-based data mining[J]. Data Mining and Knowledge Discovery, 2008, 17 (2): 129-135.
- [2] XIAO Y, WANG H, ZHANG L, et al. Two methods of selecting Gaussian kernel parameters for one-class SVM and their application to fault detection [J]. Knowledge-Based System, 2014, 59(2): 75-84.
- [3] 张金伟. 不平衡数据分类研究及在肿瘤识别中的应用[D]. 杭州:中国计量学院, 2012.
- [4] KUBAT M, HOLTE R C, MATWIN S. Machine learning for the detection of oil spills in satellite radar images[J]. Machine Learning, 1998, 30(2): 195-215.
- [5] Gu Q, Cai Z, Zhu L, et al. Data mining on imbalanced data sets[C]//Int Conf on advanced Computer Theory and Engineering. Phuket: IEEE, 2008: 1020-1024.
- [6] JASON V H, TAGHI K. Knowledge discovery from imbalanced and noisy data[J]. Data Knowledge Engineering, 2009, 68(12): 1513-1542.
- [7] 杨智明, 乔立岩, 彭喜元. 基于改进 SMOTE 的不平衡数据挖掘方法研究[J]. 电子学报, 2007, 35 (12): 22-26.
- [8] 王璐林. 面向不均衡样本的 Boosting 分类算法研究[D]. 哈尔滨:哈尔滨工业大学, 2013.
- [9] Cheng X F, Li J, Li X F. An imbalanced data classification algorithm based on under sampling[J]. Computer Engineering, 2011, 37(13): 147-149.
- [10] CHARTE F, RIVERA A J, JESUS M J D, et al. Addressing imbalance in multilabel classification: Measures and random resampling algorithms[J]. Neurocomputing, 2015(163): 3-16.
- [11] CHAWLA N V, BOWYER K W, HALL L O. SMOTE: synthetic minority over-sampling technique [J]. Journal of Artificial Intelligence Research, 2002 (16): 321-357.
- [12] HAN H, WANG W Y, MAO B H. Borderline-SMOTE: a new oversampling method in imbalanced data sets learning[C]// proc of International Conference on Intelligent Computing. Hefei, China, 2005: 878-887.
- [13] DOUZAS G, BACAO F, LAST F. Improving Imbalanced Learning Through a Heuristic Oversampling Method Based on K-Means and SMOTE[J]. Information Sciences, 2018, 465: S0020025518304997-.
- [14] NEKOOEIMEHR I, LAI-YUEN S K. Adaptive semi-supervised weighted oversampling (a-suwo) for imbalanced data sets[J]. Expert systems with applications, 2016(46): 405-416.
- [15] DOUZAS G, BACAO F. Self-Organizing Map Oversampling (SOMO) for imbalanced data set learning[J]. Expert Systems with Applications, 2017(82): 40-52.

作者简介:

柳毅 男, (1976-), 博士, 教授. 研究方向为网络安全.

曾昊(通讯作者) 男, (1993-), 硕士. 研究方向为网络安全. E-mail: 2966739148@qq.com.