第 38 卷 第 4 期 计 算 机 工 程 2012 年 2 月
Vol.38 No.4 Computer Engineering February 2012

·软件技术与数据库· 文章编号：1000—3428(2012)04—0067—03 文献标识码：A 中图分类号：TP311.13

# 一种改进的少数类样本过抽样算法

许丹丹[1]，蔡立军[1]，王 勇[2]

(1. 西北工业大学理学院，西安 710129；2. 西北工业大学计算机学院，西安 710072)

**摘 要：**针对偏斜数据集的分类问题，提出一种改进的少数类样本过抽样算法(B-ISMOTE)。在边界少数类实例及其最近邻实例构成的 $n$ 维球体空间内进行随机插值，以此产生虚拟少数类实例，减小数据的不均衡程度。在实际数据集上进行实验，结果证明，与 SMOTE 算法和 B-SMOTE 算法相比，B-ISMOTE 算法具有较优的分类性能。

**关键词：**偏斜数据集；分类；过抽样；虚拟实例；$n$ 维球体空间

# Improved Over-sampling Algorithm of Minority Class Sample

XU Dan-dan[1], CAI Li-jun[1], WANG Yong[2]

(1.School of Science, Northwestern Polytechnical University, Xi'an 710129, China;
2. School of Computer, Northwestern Polytechnical University, Xi'an 710072, China)

**【Abstract】**Aiming at the classification of the skewed dataset, this paper proposes an improved over-sampling algorithm of minority class sample, named B-ISMOTE. It improves the data unbalanced distribution of degree through randomized interpolation to produce virtual minority class instances in the sphere space, which constitute of the borderline minority class instances and its nearest neighbor. Experimental results on the real datasets show that compared with SMOTE algorithm and B-SMOTE algorithm, B-ISMOTE algorithm has better classification performance.

**【Key words】**skewed dataset; classification; over-sampling; virtual instance; $n$ dimension sphere space

**DOI:** 10.3969/j.issn.1000-3428.2012.04.022

## 1 概述

偏斜数据挖掘可应用于风险管理、网络入侵检测、信用卡欺诈检测和卫星图像中石油泄漏检测等领域。这些领域中的数据分布是不均衡的，如网络日志中“入侵行为”通常少于“正常访问”，信用卡使用中的“欺诈行为”通常少于“正常使用”。由于数据的这些特点，传统的机器学习分类算法不再适用，因此有必要寻求一种新的分类算法，使其能在数据分布不均衡的条件下，对少数类、多数类进行准确分类。

目前，研究者已经提出了一些解决数据分布不均衡问题的方法。这些方法可以分为 2 类[1]：数据水平方法和算法水平方法。数据水平方法通过重抽样来均衡数据集，包括对少数类实例进行过抽样(Over-sampling)[2]和对多数类实例进行欠抽样(Under-sampling)[3-4]。使用过抽样和欠抽样均可以减小数据的不均衡程度，但仍存在一些弊端，如欠抽样常常会丢失一些有用的多数类实例信息，过抽样可能会增大过分拟合的可能性。文献[5]提出 SMOTE 过抽样算法，在一定程度上减小了过分拟合程度，但这种方法认为所有少数类实例对分类的贡献是一样的，而在实际中，边界少数类实例比其他少数类实例贡献更大。针对该问题，文献[6]提出 B-SMOTE 过抽样方法，对边界少数类实例及其近邻进行了处理，在很大程度上减小了过分拟合程度，但是这种方法采用的线性插值仍有一定缺陷。本文主要针对数据水平方法进行研究，提出一种数据分布不均衡条件下的 B-ISMOTE 过抽样方法。

## 2 改进的少数类样本过抽样算法

偏斜数据集 $S=\{(X_1,y_1),(X_2,y_2),\cdots,(X_m,y_m)\}$，其中，$X_i$ 为 $S$ 中的实例，$y_i\in\{1.0,0.0\}$，$1\leqslant i\leqslant m$ 为 $X_i$ 的类标签。$S=P\cup Q$，$P$ 是少数类实例的集合，实例的类标签 $y_j=1.0$，$1\leqslant j\leqslant numP$，$numP=|P|$，$Q$ 是多数类实例的集合，实例的类标签 $y_l=0.0$，$1\leqslant l\leqslant numQ$，$numQ=|Q|$。由于偏斜数据集中数据分布是不均衡的，因此在 $S$ 中，$numP<<numQ$。

本文提出的 B-ISMOTE 过抽样算法主要思想如下：

(1)对每个少数类实例，求它的 $k$ 最近邻，通过比较近邻中少数类、多数类实例个数得到边界少数类实例。

(2)对每个边界少数类实例，分别求它的少数类最近邻和多数类最近邻。

(3)以边界少数类实例为中心，以它到其最近邻欧式距离的适当倍数 $d(0<d<1)$ 为半径的 $n$ 维球体内，随机产生虚拟少数类实例。

(4)对产生的虚拟实例进行有效性验证。

这里，不同边界少数类实例的不同类近邻对应的 $d$ 值可能不同。本文使用二分法得到虚拟少数类实例的空间半径。

图 1 给出了目标概念和偏斜数据集的分布情况，少数类实例个数远少于多数类。B-SMOTE 算法与 B-ISMOTE 算法产生虚拟少数类实例的范围如图 2 所示。图 1、图 2 中的直线是 $n$ 维超平面，表示目标概念，在实际应用中一般是未知的。分布在目标概念 *hyperplane* 上方的“+”表示少数类实例，下方的“–”表示多数类实例，圆表示 $n$ 维球体。在图 2 中，圆内线段 2 个端点分别表示边界少数类实例和它的最近邻，“[+]”表示生成的少数类噪音实例。

**基金项目：**国家自然科学基金资助项目(60873196)

**作者简介：**许丹丹(1984－)，女，硕士研究生，主研方向：偏斜数据挖掘；蔡立军、王 勇，副教授

**收稿日期：**2011-07-18 **E-mail：**xudandan@mail.nwpu.edu.cn

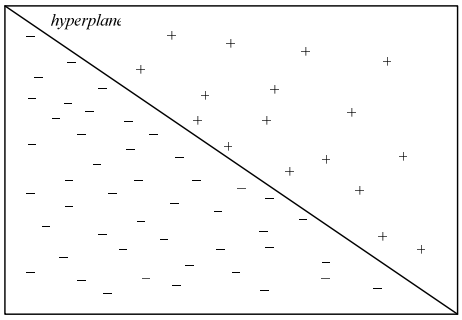


**图 1 目标概念及偏斜数据分布情况**

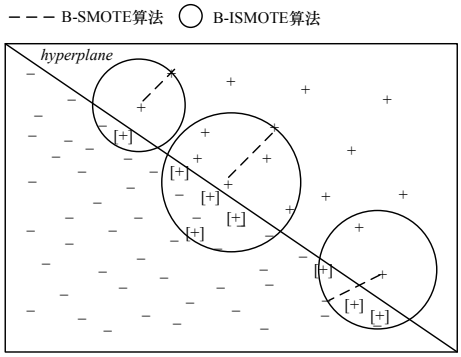


**图 2 2 种算法产生虚拟少数类实例范围的比较**

令 $X_j$ 为少数类实例，$X_{\text{nearest}}$ 为 $X_j$ 的 $k$ 个近邻中任意一个。令实例的属性个数为 $N$，那么实例 $X_j$ 可由 $X_j=(x_{j1},x_{j2},\cdots,x_{jN})$ 来表示，$x_{j1},x_{j2},\cdots,x_{jN}$ 为 $X_j$ 的 $N$ 个属性值，从而 $X_{\text{nearest}}=(x_{\text{nearest}1},x_{\text{nearest}2},\cdots,x_{\text{nearest}N})$，$X_{\text{new}}=(x_{\text{new}1},x_{\text{new}2},\cdots,x_{\text{new}N})$。B-ISMOTE 算法利用 $X_{\text{nearest}}$ 和 $X_j$ 来产生虚拟实例 $X_{\text{new}}$，$X_{\text{new}}$ 分布在图 2 中的 $n$ 维球体内，它必须同时满足下面的式子：

$$\left\|X_{\text{new}}-X_j\right\|\leqslant d_j\left\|X_j-X_{\text{nearest}}\right\| \tag{1}$$

$$x_{\text{new}i}=x_{ji}+random(0,1)\times d_j\times(b_i-a_i),1\leqslant i\leqslant N \tag{2}$$

$$a_i=x_{ji}-\left|x_{\text{nearest}i}-x_{ji}\right|,b_i=x_{ji}+\left|x_{\text{nearest}i}-x_{ji}\right|,\ 1\leqslant i\leqslant N \tag{3}$$

其中，$x_{\text{new}i}$ 是对应属性的随机值；$\left\|X_{\text{new}}-X_j\right\|$ 和 $\left\|X_j-X_{\text{nearest}}\right\|$ 分别表示对应实例间的欧式距离；$\left|x_{\text{nearest}i}-x_{ji}\right|$ 表示实例 $X_{\text{nearest}}$ 与实例 $X_j$ 对应属性差的绝对值；$d_j\times\left\|X_j-X_{\text{nearest}}\right\|$ 是适合于不同边界少数类实例的不同类最近邻的最佳半径。B-SMOTE 算法产生的 $X_{\text{new}}$ 分布于边界少数类实例及其近邻之间。B-ISMOTE 算法不但增大了 B-SMOTE 算法产生的虚拟少数类实例 $X_{\text{new}}$ 的分布范围，而且通过调整 $n$ 维球体的半径减少了虚拟实例中的噪音。

B-ISMOTE 算法的具体描述如下：

**输入** $S=\{(X_1,y_1),(X_2,y_2),\cdots,(X_m,y_m)\}=P\cup Q$
$k$=5 //最近邻实例的个数

**输出** $S^*$//均衡的数据集

1：for each example $X_i\in P$, i=1,2,…, numP;
2： $D_i$=NearestExample($X_i$, S, k);
3： if (numP < numQ), $X_i\in D$;
4：end for;
5：for each $X_j\in D$, do
6： $D_j^P$ = NearestExample($X_j$, P, k);
$D_j^Q$ = NearestExample($X_j$, Q, k);
7： $X^P_{\text{nearest,j}}$=RandomSampling( $D_j^P$ );
$X^Q_{\text{nearest,j}}$=RandomSampling( $D_j^Q$ );
8： $X_{\text{new}}$ =CreatNewExample($X_j$,$X_{\text{nearest,j}}$, $d_j$*R);
9： $X_{\text{newNearest}}$= NearestExample($X_{\text{new}}$, S, 1);
10： if $y_{\text{newNearest}}$=1.0, $X_{\text{new}}\in P_{\text{creat}}$;
11： end if;
12：end for;
13：if (num$P_{\text{creat}}$ < (numP-numQ)), return step5,
until num$P_{\text{creat}}$ = (numP-numQ);
14：$S^*=P\cup Q\cup P_{\text{creat}}$;
15：return $S^*$;

在 B-ISMOTE 算法中，第 1 行~第 4 行在 $S$ 内求得边界少数类实例；第 5 行~第 8 行在边界少数类实例及其近邻构成的 $n$ 维球体内产生虚拟少数类实例；第 9 行~第 12 行使用最近邻思想对产生的虚拟实例进行有效性验证；第 13 行~第 15 行是根据 $S$ 中多数类、少数类的实例个数得到分布均衡的数据集 $S^*$。第 8 行中产生的虚拟实例 $X_{\text{new}}$ 分布在以原少数类实例 $X_j$ 为中心，以它到 $X_{\text{nearest},j}$ 的欧氏距离 $R$ 的 $d_j$ 倍为半径的 $n$ 维球体内。不同边界少数类实例的不同类近邻所对应的 $d_j$ $(0<d_j<1)$ 值可能是不同的，可通过二分法迭代求得比较精确的 $d_j$ 值。随着迭代次数的不断增加，$d_j$ 的值越来越精确，但是需要耗费的资源就越来越多，本文限定迭代次数 $iter\leqslant 10$。下面分别给出 $X_{\text{nearest},j}$ 属于多数类和少数类时，求解 $d_j$ 的二分迭代过程。

(1)当 $X_{\text{nearest},j}$ 属于多数类时，令 $d_{j,0}=1/2$。当 $iter=t(1\leqslant t\leqslant 10)$ 时，利用式(1)~式(3)产生实例 $X_{\text{test},j}$，计算 $X_{\text{test},j}$ 在 $S$ 中的 1-最近邻。如果这个最近邻属于少数类，$d_{j,iter}=d_{j,(iter-1)}+(1/2)^{t+1}$；否则，$d_{j,iter}=d_{j,(iter-1)}-(1/2)^{t+1}$。

(2)当 $X_{\text{nearest},j}$ 属于少数类时，利用式(1)~式(3)产生虚拟实例 $X_{\text{new}}$，并求它在 $S$ 中的 1-最近邻。最近邻属于少数类时，令 $d_{j,iter}=1$；否则，$d_{j,iter}=1-\frac{1}{2^1}$。重复以上过程，直到虚拟实例的 1-最近邻属于少数类。

## 3 实验结果与分析

为了验证 B-ISMOTE 算法处理不均衡数据集的有效性，在 UCI 实际数据集上与 B-SMOTE 算法、SMOTE 算法进行比较。本文实验所采用的分类器为 C4.5、NaiveBayes 和 IBk 分类器。

### 3.1 实际数据集 UCI

本文实验所用的实际数据集是在研究不平衡数据分类时常用的 4 个公开数据集：Breast-w，Glass，Pima，Vehicle。它们都是从 UCI 的机器学习数据库[4,7-9]中获得的。其中，Breast-w 和 Pima 是 2 类数据集；Glass 和 Vehicle 是多类数据集。Glass 中含有 7 个类，将第 2 类看作少数类，其他类合并为多数类。Vehicle 中含有 4 个类，将第 1 类作为少数类，其他类合并为多数类。表 1 为 4 个实际数据集的基本信息。

**表 1 数据集的基本信息**

| 数据集 | 实例总数 | 少数类实例数 | 多数类实例数 | 少数类所占比例/(%) | 实例属性数 |
|---|---|---|---|---|---|
| Breast-w | 699 | 241 | 458 | 34.48 | 9 |
| Glass | 768 | 268 | 500 | 34.89 | 8 |
| Pima | 214 | 76 | 138 | 35.50 | 9 |
| Vehicle | 846 | 212 | 634 | 25.06 | 18 |

### 3.2 评价函数

在一般情况下，研究者采用全体实例的分类精度来评价分类器的分类性能。由于偏斜数据的分布特点使分类精度不

能很好地反映分类器的分类情况，因此本文使用 Gmean[6,10] 来评价分类器的分类性能。$acc^+$、$acc^-$ 分别表示测试集合中少数类实例的分类精确度和多数类实例的分类精确度，Gmean 值为 $\sqrt{acc^+ \times acc^-}$。要想获得较高的 Gmean 值，必须保证多数类和少数类的分类精度都很高，并且保证 2 类实例的分布是均匀的。

### 3.3　实验结果

本文采用交叉验证(Cross-validation)[7]的方法，将所有实例随机分为 5 份，并使每份样本的不平衡率保持与整体相等。然后每次取其中 4 份作为训练集，剩下的 1 份作为测试集，计算测试集上的 Gmean 值，把 5 次 Gmean 值的平均值作为该算法在整个数据集中的 Gmean 值。表 2 给出了使用 SMOTE(S)、B-SMOTE(B-S)和 B-ISMOTE(B-IS)算法处理不均衡数据集的分类结果。

**表 2　3 种算法的 Gmean 值比较**

| 数据集名称 | C4.5 分类器 | | | NaiveBayes 分类器 | | | IBk 分类器 | | |
|---|---|---|---|---|---|---|---|---|---|
| | S | B-S | B-IS | S | B-S | B-IS | S | B-S | B-IS |
| Breast-w | 0.952 4 | 0.960 7 | 0.971 4 | 0.956 5 | 0.964 1 | 0.964 1 | 0.942 4 | 0.952 0 | 0.951 4 |
| Glass | 0.740 6 | 0.729 3 | 0.765 7 | 0.612 4 | 0.593 2 | 0.626 7 | 0.755 2 | 0.781 2 | 0.786 5 |
| Pima | 0.703 4 | 0.704 9 | 0.723 7 | 0.728 0 | 0.731 5 | 0.763 7 | 0.675 1 | 0.686 1 | 0.691 4 |
| Vehicle | 0.658 5 | 0.737 1 | 0.767 5 | 0.629 3 | 0.663 2 | 0.689 7 | 0.650 2 | 0.686 3 | 0.698 2 |

Gmean 值可从整体上评价 SMOTE 算法、B-SMOTE 算法和 B-ISMOTE 算法对不同分类器分类性能的影响。在表 2 中，B-ISMOTE 算法在不同分类器、不同数据集上的 Gmean 值均大于使用 SMOTE 算法和 B-SMOTE 算法的 Gmean 值，分类器的分类性能都有不同程度的改进。与 SMOTE 算法相比，B-ISMOTE 算法不仅增大了虚拟少数类实例的生成范围，而且利用了少数类中对分类有更大影响作用的边界少数类实例，从而使得该算法的分类结果要优于 SMOTE 算法。与 B-SMOTE 算法相比，B-ISMOTE 算法除了增大了生成虚拟少数类实例的范围，还减小了噪音实例对分类的不良影响，从而使得 B-ISMOTE 算法能够更好地处理不均衡数据。由此可知，B-ISMOTE 算法优于 SMOTE 算法和 B-SMOTE 算法，它可以有效改善偏斜数据的分布不均衡程度，提高分类器的分类性能。

## 4　结束语

针对偏斜数据集，本文从数据水平的过抽样角度出发，提出一种少数类样本过抽样算法。该算法是对 B-SMOTE 算法的改进，增大了生成虚拟少数类实例的范围，同时，通过调整 $n$ 维球体的半径减小了虚拟实例中噪音的干扰。与 SMOTE 算法和 B-SMOTE 算法的比较结果证明，本文算法性能较优，能解决过分拟合问题，减小偏斜数据集的不均衡程度，提高分类器分类性能。下一步的研究目标是将特征选择与抽样算法相结合，处理不均衡数据，并寻找新算法删除噪音实例，从而减少它对分类的影响。

## 参考文献


[1] Kotsiantis S, Kanellopoulos D, Pintelas P. Handling Imbalanced Datasets: A Review[J]. GESTS International Trans. on Computer Science and Engineering, 2006, 30(1): 25-36.

[2] 杨　永, 王莉利. 基于 K-means 聚类和遗传算法的少数类样本采样方法研究[J]. 科学技术与工程, 2010, 10(10): 2334-2338.

[3] Gao Jing, Fan Wei, Han Jiawei, et al. A General Framework for Mining Concept-drifting Data Streams with Skewed Distributions[C]//Proc. of SDM'07. Minneapolis, USA: [s. n.], 2007.

[4] Gao Jing, Ding Bolin, Han Jiawei, et al. Classifying Data Streams with Skewed Class Distributions and Concept Drifts[J]. IEEE Internet Computing, 2008, 12(6): 37-49.

[5] Chawla N, Bowyer K, Hall L, et al. SMOTE: Synthetic Minority Over-sampling Technique[J]. Journal of Artificial Intelligence Research, 2002, 16: 321-357.

[6] Han Hui, Wang Wenyuan, Mao Binghuan. Borderline-SMOTE: A New Over-sampling Method in Imbalanced Data Sets Learning[C]//Proc. of IEEE International Conference on Intelligent Computing. Hefei, China: [S. l.], 2005.

[7] 王和勇, 樊泓坤, 姚正安. SMOTE 和 Biased-SVM 相结合的不平衡数据分类方法[J]. 计算机科学, 2008, 35(5): 174-176.

[8] 韩　慧, 王　路, 温　明, 等. 不均衡数据集学习中基于初分类的过抽样算法[J]. 计算机应用, 2006, 26(8): 1894-1897.

[9] 杜　娟, 衣治安, 周　颖. 基于聚类和遗传交叉的少数类样本生成方法[J]. 计算机工程, 2009, 35(22): 182-184.

[10] Kang P, Cho S. EUS SVMs: Ensemble of Under-sampled SVMs for Data Imbalance Problems[C]//Proc. of ICONIP'06. Hong Kong, China: [s. n.], 2006.


编辑　顾姣健

(上接第 66 页)


[4] Braam P J. Lustre: A Scalable, High-performance File System[EB/OL]. (2002-05-06). ftp://ftp.uni-duisburg.de/linux/filesys/Lustre/whitepaper.pdf.

[5] Rodeh O, Teperman A. zFS——A Scalable Distributed File System Using Object Disks[C]//Proc. of the 11th NASA Goddard Conference on Mass Storage Systems and Technologies. San Diego, USA: [s. n.], 2003: 207-218.

[6] Brandt S A, Xue Lan, Miller E L, et al. Efficient Metadata Management in Large Distributed File Systems[C]//Proc. of the 11th NASA Goddard Conf. on Mass Storage Systems and Technologies. San Diego, USA: IEEE Computer Society, 2003: 290-298.

[7] 吴　伟, 谢长生, 韩德志, 等. 海量存储系统中高可扩展性元数据服务器集群设计[J]. 计算机科学, 2007, 34(7): 106-109.

[8] 黄　华. 蓝鲸分布式文件系统元数据服务[J]. 计算机工程, 2008, 34(7): 4-9.

[9] 刘　仲, 周兴铭. 基于目录路径的元数据管理方法[J]. 软件学报, 2007, 18(2): 236-245.

[10] Weil S A, Pollack K T, Brandt S A, et al. Dynamic Metadata Management for Petabyte-scale File Systems[C]//Proc. of ACM/IEEE Conference on Supercomputing. Washington D. C., USA: IEEE Computer Society, 2004: 4-15.

[11] 张敬亮, 张军伟, 张建刚, 等. 蓝鲸文件系统中元数据与数据隔离技术[J]. 计算机工程, 2010, 36(2): 28-30.


编辑　任吉慧