◎理论与研发◎

基于冗余度的*K*NN训练样本裁剪新算法

王子旗,何锦雯,蒋良孝
中国地质大学(武汉) 计算机学院,武汉 430074

摘 要:作为数据挖掘领域十大算法之一,*K*-近邻算法(*K*-Nearest-Neighbor,*K*NN)因具有非参数、无需训练时间、简单有效等特点而得到广泛应用。然而,*K*NN算法在面对高维的大训练样本集时,分类时间复杂度高的问题成为其应用的瓶颈。另外,因训练样本的类分布不均匀而导致的类不平衡问题也会影响其分类性能。针对这两个问题,提出了一种基于冗余度的*K*NN分类器训练样本裁剪新算法(简记为RB*K*NN)。RB*K*NN通过引入训练样本集预处理过程,对每个训练样本进行冗余度计算,并随机裁剪掉部分高冗余度的训练样本,从而达到减小训练样本规模、均衡样本分布的目的。实验结果表明,RB*K*NN可在保持或改善分类精度的前提下显著提升*K*NN的分类效率。

关键词:*K*NN分类器;样本裁剪;快速分类;类不平衡
文献标志码:A **中图分类号:**TP391 doi:10.3778/j.issn.1002-8331.1809-0275

王子旗,何锦雯,蒋良孝.基于冗余度的*K*NN训练样本裁剪新算法.计算机工程与应用,2019,55(22):40-45.
WANG Ziqi, HE Jinwen, JIANG Liangxiao. New redundancy-based algorithm for reducing amount of training examples in *K*NN. Computer Engineering and Applications, 2019, 55(22):40-45.

New Redundancy-Based Algorithm for Reducing Amount of Training Examples in *K*NN

WANG Ziqi, HE Jinwen, JIANG Liangxiao
School of Computer Science, China University of Geosciences, Wuhan 430074, China

Abstract: As one of the top 10 algorithms in data mining, the *K*-Nearest-Neighbor(*K*NN) algorithm is widely used because it is an non-parametric, simple and effective algorithm without training time. However, when it faces to massive amount of high-dimensional training examples, its high classification time complexity becomes a bottleneck of its application. In addition, its classification performance is often harmed, when the class distribution of training examples is skewed and the class imbalance problem occurs. To address these two issues, this paper proposes a new redundancy-based algorithm for reducing the amount of training examples (simply RB*K*NN). RB*K*NN at first computes the redundancy of each training example, and then randomly deletes some high redundant training examples by introducing a pre-processing process. RB*K*NN can not only reduce the size of training example set, but also make the class distribution of training examples more balanced. The experimental results show that RB*K*NN significantly promotes the efficiency of *K*NN, yet at the same time maintains or improves the classification accuracy of *K*NN.

Key words: *K*NN classifiers; example reduction; fast classification; class imbalance

1 引言

随着信息时代的迅猛发展,海量数据生成、积累,对数据进行快速分类、处理显得尤为关键。数据挖掘技术在大数据的背景下应运而生。在现有的数据挖掘技术中,常见的分类方法包括支持向量机[1]、决策树[2]、贝叶斯网络[3]、人工神经网络[4]、*K*-近邻算法(*K*-Nearest-Neighbor,*K*NN)[5]等。其中,*K*NN作为一种非参数、无需训练时间、简单高效的算法,最初被用于解决文本分

基金项目:国家自然科学基金联合基金重点项目(No.U1711267)。
作者简介:王子旗(1996—),男,研究生,主要研究方向为机器学习与数据挖掘;何锦雯(1997—),女,研究生,主要研究方向为机器学习与数据挖掘;蒋良孝(1977—),男,教授,主要研究方向为机器学习与数据挖掘,E-mail:ljiang@cug.edu.cn。
收稿日期:2018-09-21 **修回日期:**2018-11-21 **文章编号:**1002-8331(2019)22-0040-06
CNKI**网络出版:**2019-01-15, http://kns.cnki.net/kcms/detail/11.2127.TP.20190114.1717.010.html

类问题，后来被广泛应用于模式识别的各个领域，并且取得了很好的效果。

KNN分类器的基本思想是寻找在特征空间中与待测样本特征距离最小的k个训练样本，并将待测样本最终分类到k个训练样本中具有优势数量的类中。然而，作为一种基于实例的懒惰学习方法，KNN分类器需要存储全部训练样本，并且在分类时需要计算待测样本与全部训练样本之间的特征距离并按距离进行排序，这意味着KNN分类器的分类时间复杂度将随着训练样本数或特征维度的增加而急剧上升。另外，因训练样本的类分布不均匀而导致的类不平衡问题也会影响分类性能。

目前，针对KNN分类效率的改进方法大致可以分为两类：第一类是采用快速搜索方法，通过提高搜索速度直接提高KNN分类效率[6-7]，如Zhong的G-Tree算法[8-9]、Deng的基于聚类加速的KNN改进算法[10]和Xie的Simba（Spatial In-Memory Big Data Analysis）算法[11]。另一类是精简训练样本数量或特征维度。精简特征维度主要采用特征选择或特征抽取的方式。精简训练样本数量主要通过样本裁剪，将对分类效果影响不大的训练样本从训练集中去除，只保留更具代表性的样本。本文主要讨论通过样本裁剪精简训练样本数量以提高KNN分类器效率的方法。经典的KNN训练样本集裁剪算法主要有Hart的Condensing算法[12]、Wilson的Editing算法[13]以及DROP（Decremental Reduction Optimization Procedure）算法[14]和Devijver的MultiEdit算法[15]等。此类经典裁剪算法提出的时间较早，具有诸多弊端，例如需要不停迭代，直到训练集不再变化，这导致训练集过大时预处理的时间开销很大。同时，这些方法也未考虑类不平衡问题对KNN分类性能的影响。

除了上述算法以外，李荣陆和胡运发[16]提出了一种基于密度的KNN文本分类器训练样本裁剪算法（Density-Based method for reducing the amount of training data in KNN text classification，DBKNN）。该算法的优势在于无需反复迭代，而是基于样本分布密度进行裁剪，以使得训练样本的分布尽量均匀。然而在裁剪过程中，该算法并未考虑被裁剪样本的类标记，这导致在密度衡量过程中所有训练样本的类分布信息被忽略，最终得到的密度信息只包含区域密度而丢失了类密度信息。只考虑区域密度而不考虑类密度会导致高密度区域中的稀有类样本面临被裁剪的风险，最终使得稀有类样本更加稀少。

本文提出了一种基于冗余度的KNN分类器训练样本裁剪新算法（RBKNN），将训练样本区域同类密度作为冗余度，并对高冗余度样本进行裁剪，使得训练样本数量大幅减少，类的分布更加均衡。此方法只需两次遍历训练样本集即可完成对训练样本的裁剪，克服了已有经典裁剪算法的缺陷。同时，兼顾区域密度与类密度，保证了KNN分类器的分类性能。

2 样本冗余度与区域同类样本密度

KNN分类器包含三种思想："众数思想"、"代表思想"和"区域思想"。"众数思想"，即k个最近邻样本的类标记的众数决定了待测样本的类标记，这意味着只要保证k个最近邻样本的类标记的众数不变，各类样本数量的小幅度变化并不会影响分类的结果。"代表思想"，即k个最近邻样本具有代表性，通过选出与待测样本相似且最具代表性的训练样本来决定待测样本的类别。"区域思想"，即KNN分类器只在待测样本特征空间中一个小邻域内统计训练样本的类标记信息，这意味着与待测样本特征距离很大的训练样本不会对待测样本的分类结果产生影响。

基于上述三种思想便可做以下三点关于训练样本冗余度的推论：

首先，训练样本的冗余度体现在"众数思想"上。即当某一类别样本在某一特征空间区域内占绝大多数时，即使小幅度减少此类样本也不影响其在此区域内的数量优势。

其次，训练样本的冗余度也体现在"代表思想"上。即当某一样本在某一区域内可被越多的其他样本代替或代表，其冗余度就会越大。

最后，训练样本的冗余度还体现在"区域思想"上。即对于某个训练样本的冗余度度量应限制在某一邻域，因为此训练样本只对其小邻域内待测样本的分类结果产生影响。

基于上述关于训练样本冗余度的推论，本文将某个训练样本的冗余度定义为在其小邻域内，具有相同类标记的训练样本的密度，即区域同类密度。对于一个训练样本而言，其区域同类密度越大，此类在此区域就越具数量优势，在此区域内此类样本可被小幅度裁剪。换言之，某样本区域类密度越大，此样本可被越多的样本代替或代表，其冗余度也就越大。

3 基于冗余度的KNN训练样本裁剪算法

3.1 基本定义

训练样本裁剪方案需要解决两个问题：样本冗余度计算问题和高冗余样本裁剪问题。

针对第一个问题，上文已经提到，可以将训练样本的冗余度定义为该样本的区域同类密度。具体定义如下：

给定训练样本集$D=\{\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_l\}$，其中$\boldsymbol{X}_i\in\mathbb{R}^n$，$i=1,2,\cdots,l$。

定义1 设$Neighbor(\boldsymbol{X},k')$为训练样本$\boldsymbol{X}$在训练样本集$D$中不包含$\boldsymbol{X}$本身的$k'$个最近邻样本；$k'$为冗余度度量范围；$Class(\boldsymbol{X})$为样本$\boldsymbol{X}$的类别。则定义样本$\boldsymbol{X}$的区域同类样本数为：

$$RsNum(\boldsymbol{X},k')=\left|\{\boldsymbol{Y}|\boldsymbol{Y}\in Neighbor(\boldsymbol{X},k'),Class(\boldsymbol{Y})=Class(\boldsymbol{X})\}\right|$$

此处需注意,作为确定冗余度度量范围的参数 k' 与 *K*NN 分类器中的参数 k 并不一定相同, k' 值越大,意味着将在更大范围内评估某训练样本的冗余度。

设该区域的超体积为 V ,则其区域密度为 k'/V ,类密度为 $RsNum(\boldsymbol{X},k')/V$ 。同时考虑区域密度与类密度对冗余度的影响,在此将区域同类密度 $RDensity(\boldsymbol{X},k')$ 定义为类密度与区域密度的比值:

$$RDensity(\boldsymbol{X},k')=\frac{RsNum(\boldsymbol{X},k')}{V}/\frac{k'}{V}$$

将上式化简后可得到以下定义:

定义 2 训练样本 $\boldsymbol{X}$ 的区域同类密度 $RDensity(\boldsymbol{X},k')$ 为 $\boldsymbol{X}$ 的区域同类样本数 $RsNum(\boldsymbol{X},k')$ 与度量范围 k' 的比值:

$$RDensity(\boldsymbol{X},k')=\frac{RsNum(\boldsymbol{X},k')}{k'}$$

针对第二个问题,考虑到本文的主要目标是裁剪冗余度高的训练样本。一种直接思路是对训练样本冗余度进行排序,并将冗余度最高的部分样本进行裁剪。然而,这种方法存在以下问题:在进行样本冗余度计算时,因为类中心区域样本具有较大的类密度,所以其冗余度往往高于边界区域样本冗余度。这就导致类中心区域样本会因冗余度排序高被率先全部去除,从而形成类中心区域样本空洞。虽然类中心区域的大部分样本对分类决策没有太大的影响,但全部去除也会产生一些问题。更何况在实际的训练样本中,类与类之间的边界并不是非常明显,而且会有一些交叉重叠。因此,较为合适的解决方法是适当降低类中心区域的密度,即适当保留一些处于类中心区域的训练样本以保证分类的性能。

本文拟采用基于冗余度的随机方法进行部分训练样本的裁剪,即高冗余度样本拥有更大的裁剪概率。此方法保证了即使冗余度最高的区域也不会形成样本空洞。故在此定义:

定义 3 若给定最大裁剪概率为 $MaxRmRate$ 和最小冗余度阈值为 $MinRedund$,样本的裁剪概率为:

$$specRmRate(RDensity)=\frac{MaxRmRate(RDensity-MinRedund)}{1-MinRedund}$$

此处有几点需要注意:首先,最大裁剪概率 $MaxRmRate<1$ 可以保证即使冗余度最大的区域($RDensity=1$)也会有样本保留。其次,最低冗余度阈值 $MinRedund$ 的引入解决了另一个问题,即当样本冗余度不够大(即某区域内某一类所占的数量优势并不足够突出)时,贸然裁剪样本可能导致原优势类失去数量优势,致使区域优势类改变。而 $MinRedund$ 可保证只有 $RDensity$ 足够大($RDensity>MinRedund$)时,才会开始进行裁剪,降低了发生过度裁剪的风险。

3.2 裁剪算法

Input 待裁剪训练样本集合 $\boldsymbol{D}$,冗余度评估范围 k' ,最大裁剪概率 $MaxRmRate$,最小冗余度阈值 $MinRedund$

For 每一个训练样本 $\boldsymbol{X}\in\boldsymbol{D}$ do

根据定义1和定义2,计算 $\boldsymbol{X}$ 的区域同类密度: $RDensity(\boldsymbol{X},k')$;

根据定义3,计算 $\boldsymbol{X}$ 的裁剪概率: $specRmRate(\boldsymbol{X})$;

随机产生一个0到1的随机数 $Random$

If ($specRmRate(\boldsymbol{X})>Random$ 且 $RDensity(\boldsymbol{X},k')>MinRedund$) then

将 $\boldsymbol{X}$ 从训练样本集 $\boldsymbol{D}$ 中裁剪;

Endif

Output 裁剪后的训练样本集 $\boldsymbol{D}'$

3.3 参数分析

k' :该参数用来确定样本冗余度的计算范围。例如将 k' 设置为10,表示在10个最近邻训练样本组成的邻域内进行区域同类样本密度计算(样本冗余度计算)。实验表明,此值越大样本裁剪标准越严格,换言之,样本裁剪率越低。此值一般取10~20能取得较好的裁剪效果。

$MaxRmRate$:该参数用来控制高冗余区域(例如类中心区域)的裁剪力度。此值越大,高冗余区域保留样本越少,但为防止出现高冗余区域样本空洞,此值一般小于1。实验表明其在0.6~0.8较为合理。

$MinRedund$:该参数用来防止过裁剪而导致区域样本数量失衡而限定的裁剪所需最小冗余度。一般此值越大,样本越不易失衡,但裁剪率也会更小。实验表明, $MinRedund$ 为0.9较为合适,但当训练样本分界十分明显且训练精度很高时,也可在0.5~0.9之间取值。

3.4 对类不平衡问题的优化

类不平衡问题是数据挖掘领域的常见问题之一。在分类问题中,类不平衡问题指某些类的样本要比其他类的样本多得多。当训练样本存在类不平衡问题时,*K*NN 分类器可能会将少数类中的样本错分到多数类。例如在图1中,测试点样本本应属于“×”类,但因为“×”类样本量低于“○”类样本量,所以在 k 取15时,测试点样本以○:×=9:6的投票比例被误分类到“○”类中。并且 k 值越大,类不平衡问题就越严重。

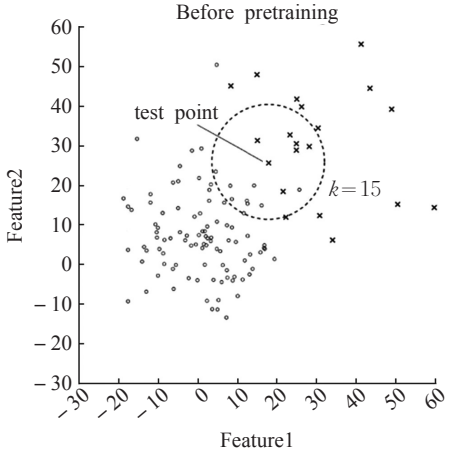

图1 类不平衡问题对分类结果的影响(裁剪前)

导致类不平衡问题的一个主要原因是训练样本的类分布不均匀，而本文所提出的基于样本冗余度的训练集裁剪算法可以较好地解决类分布不均匀的问题。例如，在图1中，若 k' 取值15，*MinRedund* 取值0.9，则可计算出测试点样本的冗余度为0.4。因为小于 *MinRedund*，所以裁剪概率为0。同理可知，处于边界位置的“×”类样本因为在类密度上并不占优势，所以裁剪概率相应较小。相反，处于边界的“○”类样本由于类密度较大，因此具有更大的裁剪概率。于是经过训练样本裁剪后，“○”类样本在边界区域分布密度将减少，而“×”类样本分布密度则基本不变，这便实现了样本分布密度的均衡化。

正如图2所示，在进行除冗余裁剪后，训练集样本分布更加均匀。此时再对测试点样本分类，若依旧取 k 为15，则会以○∶×=7∶8的比例将测试点样本正确分类到“×”类。

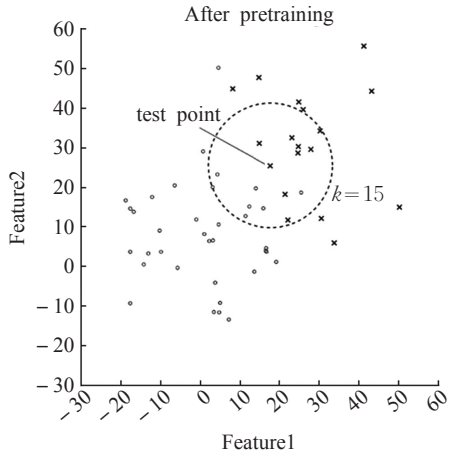

图2　类不平衡问题对分类结果的影响（裁剪后）

4　实验结果

本文设计了两组实验来分别验证RB*K*NN算法在裁剪训练样本集上的表现以及对类不平衡问题的优化效果。

在第一组实验中，将本文提出的RB*K*NN算法与经典的*K*NN算法以及李荣陆与胡运发[16]提出的DB*K*NN算法进行分类精度、裁剪率等方面的比较，在各算法上均采用10次运行10折交叉验证，并利用数据挖掘软件工具KEEL完成非参数统计方法维尔科克森符号秩检验（Wilcoxon Signed Rank Test），对各算法的分类精度和裁剪算法的裁剪率进行显著性检测。

在第二组实验中，将RB*K*NN和DB*K*NN算法[16]应用于存在类不平衡问题的手写数字识别数据集中，比较两种算法处理类不平衡问题的优化效果。为保证实验的有效性，以上算法实验环境均为MatlabR2017b、64位Windows10系统，实验中各算法均运行在相同训练数据集上，并在相同的测试数据集上进行评估。

为了检验样本裁剪算法的裁剪效果以及对*K*NN分类器分类精度的影响，第一组实验在15个标准的UCI（University of California，Irvine）分类数据集上对本文提出的RB*K*NN算法与经典*K*NN和DB*K*NN算法[16]进行了比较测试，所有算法中 k 值为5。实验中采用的15个数据集的具体信息如表1所示。

表1　UCI数据集描述

Dataset	#Examples	#Features	#Classes
balance_scale	625	4	3
banknote	1 372	4	2
breast	699	9	2
diabetes	768	8	2
glass	214	9	6
iris	150	4	3
letter	20 000	16	26
pageblocks	5 473	10	5
pima	768	8	2
segment	2 310	19	7
shuttle	14 500	9	7
spambase	4 601	57	2
vehicle	846	18	4
wdbc	569	30	2
wisconsin	699	9	4

这15个数据集覆盖了较为广泛的领域、数据特征和类型，这既保证了实验的准确性，也可证明基于冗余度的样本裁剪算法（RB*K*NN）具有较强的适应性。

实验中，DB*K*NN算法[16]的参数包括 r、*MinPts*、*LowPts*。其中 r 表示统计密度范围的半径，取值为当前数据集中所有数据的 k 个最近邻样本的距离的平均值；*MinPts* 表示最小样本数，为数据集类别平均样本数的8%；*LowPts* 始终小于 *MinPts*，用于衡量类内样本分布密度，取 *MinPts* 的0.7倍。

本文所提RB*K*NN算法的参数包括 k'、*MinRedund* 和 *MaxRmRate*。其中 k' 为裁剪算法计算样本冗余度的范围，即统计 k' 个最近邻，*MinRedund* 为样本裁剪最小冗余度，*MaxRmRate* 为最大裁剪概率。对于最大裁剪概率的最优取值，本文在15个数据集上进行实验，观察 *MaxRmRate* 取值对算法精度的影响，实验结果如图3所示。由图可知，有些数据集冗余度较小，需要裁剪的样本较少，因此最大裁剪概率取值对其影响不大，而对于冗余度较大的数据集来说，存在一个最大裁剪概率的临界点，当最大裁剪概率小于此临界点时，最大裁剪概率取值对精度影响不大；当最大裁剪概率大于此临界点时，进入过度裁剪阶段，此时分类精度将会随着最大裁剪概率的增大而大幅下降。在实验中的所有数据集最大裁剪概率临界点均大于0.7，考虑到实验数据集覆盖范围较广，认为此结果具有普适性。因此，为保持分类器分类性能，在以下实验中，最大裁剪概率 *MaxRmRate* 均取0.7。RB*K*NN算法实验参数取值如表2所示。

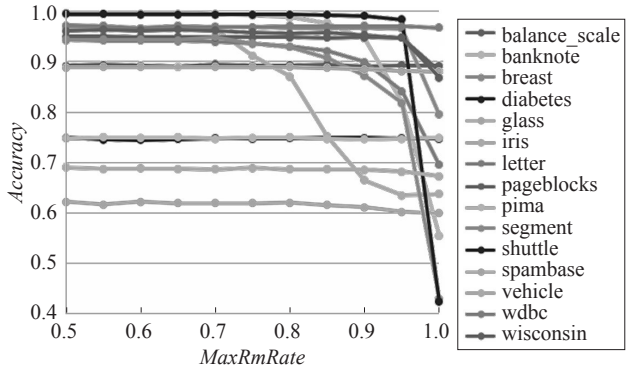

图3 最大裁剪概率*MaxRmRate*对分类精度影响

表2 RB*K*NN算法参数设置

k′	*MinRedund*	*MaxRmRate*
15	0.9	0.7

本次实验比较了本文提出的RB*K*NN算法与经典*K*NN算法和DB*K*NN算法[16]的分类精度以及RB*K*NN算法和DB*K*NN算法[16]的裁剪效果。具体的分类精度结果如表3所示,算法裁剪效果如表4所示,其中“裁剪率”=“裁剪数量”/“样本总数”。分类精度平均值以及裁剪率平均值分别列在表3和表4的底部。

表3 UCI数据集分类精度比较结果 %

Dataset	*K*NN	DB*K*NN	RB*K*NN
balance_scale	89.60	89.50	89.35
banknote	99.67	99.27	99.45
breast	96.54	96.55	96.80
diabetes	74.63	73.95	74.67
glass	61.67	55.90	62.33
iris	96.00	96.00	95.80
letter	94.20	90.76	93.97
pageblocks	95.00	94.98	94.93
pima	74.38	74.14	74.37
segment	94.40	92.68	94.14
shuttle	99.55	98.38	99.34
spambase	88.97	86.79	89.01
vehicle	68.64	68.12	69.07
wdbc	97.07	96.52	97.09
wisconsin	96.33	96.33	96.28
Average	88.44	87.32	88.44

另外,本文还在KEEL平台上应用了维尔科克森符号秩检验对各算法在15个数据集上的分类精度及裁剪算法在15个数据集上的裁剪率都进行了显著性检验,具体的测试结果如表5和表6所示。

表5和表6中,位于对角线以下区域的每一个元素为该行算法优于该列算法的秩总和,记为R^+,位于对角线以上区域的每一个元素为该列算法劣于该行算法的秩总和,记为R^-。另外,根据维尔科克森符号秩检验的临界值表,当可信度为0.05、数据集个数为15时,若R^+和R^-中的较小值小于或者等于临界值25,则认为这两种算法在分类精度(表5)或裁剪率(表6)上是有显著差异的。从实验结果可以得到以下结论:本文提出的RB*K*NN算法在不降低*K*NN分类器分类精度的前提下,可以大幅度地裁剪训练样本,从而提升*K*NN分类器的分类效率。更详细的实验结论概述如下:

表4 UCI数据集预处理裁剪率比较结果 %

Dataset	DB*K*NN	RB*K*NN
balance_scale	30.46	27.99
banknote	32.49	67.58
breast	53.55	56.74
diabetes	33.40	11.68
glass	35.74	8.02
iris	16.46	48.53
letter	23.94	39.77
pageblocks	50.97	57.85
pima	33.32	11.48
segment	49.52	69.16
shuttle	33.19	52.00
spambase	47.22	32.89
vehicle	20.61	9.82
wdbc	39.01	52.29
wisconsin	0.00	56.32
Average	33.33	40.14

表5 分类精度的维尔科克森测试秩计算结果

算法	*K*NN	DB*K*NN	RB*K*NN
*K*NN	—	115.5	69.5
DB*K*NN	4.5	—	11.0
RB*K*NN	50.5	109.0	—

表6 裁剪率的维尔科克森测试秩计算结果

算法	DB*K*NN	RB*K*NN
DB*K*NN	—	44.0
RB*K*NN	76.0	—

(1)在分类精度的比较方面,具体如表3和表5所示,本文提出的RB*K*NN算法与经典*K*NN算法在分类精度上并无显著差异,也就是说经过RB*K*NN算法裁剪的数据集不会影响*K*NN分类器的分类精度。另外,RB*K*NN算法在分类精度上显著优于DB*K*NN算法[16]。而DB*K*NN算法[16]则在分类精度上与经典*K*NN算法有显著差异。这说明经过DB*K*NN算法裁剪的数据集会使*K*NN分类器的分类精度有所下降,简言之,此算法提高*K*NN分类器的分类效率是以牺牲分类精度为代价的。

(2)在裁剪率的比较方面,具体如表4和表6所示,本文提出的RB*K*NN算法在所选用的15个数据集上的裁剪率高达8.02%~69.16%,这使得RB*K*NN分类器的分类时间较之于经典*K*NN算法也相应降低了8.02%~69.16%。且RB*K*NN的平均裁剪率高达40.14%,明显高于DB*K*NN算法[16]的平均裁剪率(33.33%),这使得RB*K*NN算法在处理大规模数据集时更显优势。另外,本文的RB*K*NN算法在原理和实现上比DB*K*NN算法[16]更简单,预处理过程中的计算量也更少,因此在实验中RB*K*NN算法可在更短的时间内完成预处理。

综上所述，本文提出的RB*K*NN算法提升*K*NN分类器的分类效率并不是以降低其分类精度为代价而取得的，在分类精度的比较结果中，个别数据集经过RB*K*NN算法处理后甚至表现出了更好的分类性能。因此，本文提出的RB*K*NN算法可以在维持分类器精度的前提下，对规模较大的数据集进行大幅度裁剪，以提高分类器的分类效率。

最后，为验证本文所提的RB*K*NN算法对类不平衡问题的优化效果，使用斯坦福大学提供的手写数字识别数据集对算法进行了进一步测试。该数据集共有5 000个样本，10个类别，每个类别500个样本。将数据集随机均分成三份，一份作为训练集，一份作为测试集，并取最后一份中全部偶数类别样本作为补充训练集添加到训练集中。这样得到的训练集中的偶数类别样本数便是奇数类别样本数的两倍，以形成类不平衡的效果。本实验比较了本文提出的RB*K*NN算法和已有的DB*K*NN算法[16]。实验中所有参数与上一组实验一致，实验结果经过5次重复测试求平均值，具体比较结果如表7和表8所示。

表7　手写数字识别数据集分类精度比较结果　%

Dataset	*K*NN	DB*K*NN	RB*K*NN
stanford_digit	92.95	92.89	93.17

表8　手写数字识别数据集预处理裁剪率比较结果　%

Dataset	DB*K*NN	RB*K*NN
stanford_digit	25.40	42.07

由实验结果可知，本文提出的RB*K*NN算法在此类不平衡数据集中的分类精度和裁剪效果均优于已有的DB*K*NN算法[16]。因此，RB*K*NN算法不仅可以提升*K*NN分类器的分类效率，还可以通过裁剪冗余样本来改善训练样本的类不平衡性，从而提高其分类精度。

5　结束语

本文针对*K*NN分类器面临的分类时间复杂度高和类不平衡问题，提出了一种基于冗余度的*K*NN分类器训练样本裁剪新算法（RB*K*NN）。新算法通过引入训练样本集预处理过程，对每个训练样本进行冗余度计算，并随机裁剪掉部分高冗余度的训练样本，从而达到减小训练样本规模、均衡样本分布的目的。大量的实验结果表明，新算法可在保持或改善分类精度的前提下，显著提升*K*NN的分类效率。同时，在面对类不平衡问题时，新算法也可以通过裁剪高冗余样本，改善训练样本的类不平衡性，从而提高其分类精度。

但在某些方面，新算法依旧存在改进的空间。例如，在冗余度计算过程中存在需要人工设定的参数；同时，在区域同类密度度量方法上只是简单地采用统计样本数目的方式，这导致计算得到的冗余度是一个离散值，在k'值过小时难以对不同训练样本的冗余度进行细分。今后，会采取距离加权等方式来计算训练样本的冗余度，使得本文所提的裁剪算法能在k'值较小时依然有效。

参考文献：

[1] Keerthi S S，Shevade S K，Bhattacharyya C，et al.Improvements to Platt's SMO algorithm for SVM classifier design[J].Neural Computation，2001，13(3)：637-649.

[2] Quinlan J R.Simplifying decision trees[J].International Journal of Man-Machine Studies，1987，27(3)：221-234.

[3] Lewis D D.Naive (Bayes) at forty：the independence assumption in information retrieval[C]//LNCS 1398：European Conference on Machine Learning，1998：4-15.

[4] Mcculloch W S.A logical calculus of ideas imminent in nervous activity[J].The Bulletin of Mathematical Biophysics，1943，5(4)：115-133.

[5] Yang Y.A re-examination of text categorization methods[C]//International Conference on Research and Development in Information Retrieval，1999：42-49.

[6] Belkasim S O，Shridhar M，Ahmadi M.Pattern classification using an efficient *K*NNR[J].Pattern Recognition，1992，25(10)：1269-1274.

[7] Vidal E.An algorithm for finding nearest neighbors in (approximately) constant average time[J].Pattern Recognition Letters，1986，4(3)：145-157.

[8] Zhong R，Li G，Tan K L.G-Tree：an efficient and scalable index for spatial search on road networks[J].IEEE Transactions on Knowledge & Data Engineering，2015，27(8)：2175-2189.

[9] Zhong R，Li G，Tan K L.G-Tree：an efficient index for *K*NN search on road networks[C]//ACM International Conference on Information & Knowledge Management，2013：39-48.

[10] Deng Z，Zhu X，Cheng D.Efficient *K*NN classification algorithm for big data[J].Neurocomputing，2016，195(3)：143-148.

[11] Xie D，Li F，Yao B.Simba：spatial in-memory big data analysis[C]//International Conference on Advances in Geographic Information Systems，2016：1071-1085.

[12] Hart P.The condensed nearest neighbor rule[J].IEEE Transactions on Information Theory，2003，14(3)：515-516.

[13] Wilson D L.Asymptotic properties of nearest neighbor rules using edited data[J].IEEE Transactions on Systems，Man and Cybernetics，1972，2(3)：408-421.

[14] Wilson D R，Martinez T R.Reduction techniques for instance based learning algorithms[J].Machine Learning，2000，38(3)：257-286.

[15] Devijver P A，Kittler J.Pattern recognition：a statistical approach[M].Upper Saddle River：Prentice Hall，1982.

[16] 李荣陆，胡运发.基于密度的*K*NN文本分类器训练样本裁剪方法[J].计算机研究与发展，2004，41(4)：539-545.