一种基于 Tri-training 的众包标记噪声纠正算法

杨 艺[1],蒋良孝[1,2],李超群[3],李宏伟[3]
(1. 中国地质大学计算机学院,湖北武汉 430074; 2. 智能地学信息处理湖北省重点实验室(中国地质大学),湖北武汉 430074;
3. 中国地质大学数学与物理学院,湖北武汉 430074)

摘 要: 在众包学习中,使用标记集成算法得到的集成标记中仍然存在一定程度的标记噪声. 本文受三重训练思想的启发,提出了一种基于 tri-training 的众包标记噪声纠正算法(Tri-Training-based Label Noise Correction,TTLNC). TTLNC 首先使用过滤器获得干净集和噪声集,然后在干净集上进行 bagging 分别训练三个不同的分类器,并通过这些分类器重新标注噪声集中的实例,同时按照实例分配策略将实例分配给相应的训练集. 最后在新训练集上重新训练三个不同的分类器,并用新分类器的分类结果重新标注所有实例. 在仿真标准数据和真实众包数据集上的实验结果表明 TTLNC 比其他四种最先进的噪声纠正算法在噪声比和模型质量两个度量指标上表现更优.

关键词: 众包学习; 三重训练; 集成标记; 标记噪声; 噪声纠正; 噪声过滤
中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112 (2021)03-0424-11
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.12263/DZXB.20200337

A Tri-training-Based Label Noise Correction Algorithm for Crowdsourcing

YANG Yi[1], JIANG Liang-xiao[1,2], LI Chao-qun[3], LI Hong-wei[3]
(1. *School of Computer Science, China University of Geosciences, Wuhan, Hubei* 430074, *China*;
2. *Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Wuhan, Hubei* 430074, *China*;
3. *School of Mathematics and Physics, China University of Geosciences, Wuhan, Hubei* 430074, *China*)

Abstract: In crowdsourcing learning, a certain level of label noise still exists in integrated labels obtained by employing ground truth inference algorithms. Inspired by the tri-training idea, this paper proposes a tri-training-based label noise correction (TTLNC) algorithm for crowdsourcing. TTLNC at first employs a filter to get a clean set and a noisy set and then trains three different classifiers from the bagged clean set. Furthermore, each instance from the noisy set is relabeled by these classifiers and assigned to the corresponding training set according to the designed instance assignment strategy. Finally, three classifiers are retrained on three new training sets and are used to relabel all instances. Experimental results on both simulated benchmark data and real-world crowdsourced data show that TTLNC significantly outperforms other four state-of-the-art noise correction algorithms in team of the noise ratio and the model quality.

Key words: crowdsourcing learning; tri-training; integrated labels; label noise; noise correction; noise filtering

1 引言

在监督学习中,往往需要大量有标记的训练实例来保证训练模型的质量. 例如,为了建立一个模型来识别图像中是否有我们所感兴趣的对象时,首先需要做的就是在合适的环境中进行拍照获取大量图像,并由专家进行标注. 然而,在实际应用中,获取大量有标记的训练实例是十分耗时且花费高昂的[1].

近年来,随着众包技术的发展,类似于 Amazon Mechanical Turk(AMT)和 CrowdFlower 的众包平台为获取大量数据标记提供了一个经济、便捷的方式. 这种在线的众包服务允许多个众包工人对每个实例进行标注,最终数据集中的每个实例都获得一个多噪声标记集. 然后根据标记集成的方法,从每个实例的多噪声标记集中推理出一个合适

收稿日期:2020-04-07;修回日期:2020-10-28;责任编辑:覃怀银
基金项目:国家自然科学基金联合基金(No. U1711267);中央高校基本科研业务费专项资金(No. CUGGC03)

的集成标记作为该实例的标记. 目前,已经有研究者在标记集成算法的研究上做了大量工作,例如:Majority Voting (MV)[2]、ZenCrowd[3]、FaitCrowd[4]、Ground Truth Inference using Clustering(GTIC)[5]、Iterative Weighted Majority Voting (IWMV)[6]、Bi-Layer Clustering (BLC)[7]、Multiple Noisy Label Distribution Propagation (MNLDP)[8]、MV-Freq、MV-Beta、Paired-Freq 以及 Paired-Beta [9]等等.

然而,由于众包工人并非专业领域人员,不具备相关专业知识,因此提供的标记往往是不可靠的. 所以,无论通过何种标记集成算法得到实例的集成标记,其中仍含有一定程度的标记噪声. 这里说的噪声是指实例的集成标记与真实标记不一致. 显然,这种标记噪声的存在会损害训练数据的质量和后续训练模型的性能以及在现实场景中的应用效果.

为了降低标记噪声的影响,提高标记集成后的数据标记质量,在众包学习领域中标记噪声纠正算法的研究被提出. 目前,已经有研究者在标记噪声纠正方面做了一些工作. 例如,Nicholson 等人[10]提出了三种标记噪声纠正算法:Polishing Labels(PL)、Self-Training Correction(STC)和 Cluster-based Correction(CC). PL 算法对数据集使用十折交叉构建分类器,然后对每个样本进行预测,基于多数投票机制进行纠正处理. STC 算法则是在干净集上训练分类器对噪声集中实例进行纠正,纠正后的实例加入干净集,然后循环迭代. 与前两种算法不同,CC 算法是一种基于聚类的噪声纠正方法,其核心思想是在数据集上多次构建聚类,根据每次实例在簇中的分布情况对其进行纠正. 以上三种标记噪声纠正算法都是为了适应众包学习,由机器学习领域传统的噪声纠正技术改造适应而来的,都取得了非常不错的效果. 最近,有部分学专门针对众包场景下众包数据提供的信息,设计了一些新的标记噪声纠正算法,以提高众包数据和训练模型的质量,代表性算法包括 Adaptive Voting Noise Correction Algorithm(AVNC)[11]和 Between-class Margin-Based Noise Correction(BMNC)[12]. AVNC 算法提出了众包领域的噪声纠正框架,利用实例的集成标记质量和工人质量来过滤噪声实例,并基于过滤过程中的干净集构建分类器来对噪声样本进行纠正. BMNC 算法则是利用众包数据中实例的多噪声标记集去评估集成标记的置信程度,从而实现对数据集更加精准的过滤,最终用得到的干净集训练分类器对过滤出的噪声集合进行纠正.

上述的标记噪声纠正算法都是基于干净集来构建分类器,然而由于噪声过滤器的局限性,干净集中仍存在一定比例的噪声实例,因此训练出的分类器的效果难以保证,从而导致使用该分类器纠正噪声实例的结果存在较大误差. 在本文中,我们提出一种基于 Tri-training 的标记噪声纠正算法(Tri-Training-Based Label Noise Correction,TTLNC). 受半监督学习领域中三重训练[13]的启发,TTLNC 算法首先使用过滤器将数据集分为干净集和噪声集. 然后在干净集上训练三个不同分类器对噪声集实例进行重新标注,并通过本文提出的实例分配策略,将标注后的实例分配给相应的训练集,然后重新训练这三个分类器,再对整个数据集中的实例进行重新标注. 我们同时使用模拟的标准数据和真实的众包数据进行了实证研究,结果表明在噪声比和模型质量两个度量指标上,TTLNC 算法比其他四种最先进的噪声纠正算法表现更好.

2　基于 Tri-training 的标记噪声纠正算法

2.1　背景知识

我们首先定义一个众包系统,在这个系统中包含 N 个实例以及 J 个众包工人. 我们从众包系统中获得众包数据集 $D=\{(x_i,L_i)\}_{i=1}^{N}$. 其中每个实例 x_i 带有一个多噪声标记集 $L_i=\{l_{ij}\}_{j=1}^{J}$,其中 l_{ij} 表示实例 x_i 由众包工人 $u_j(j=1,2,\cdots,J)$ 给出的标记. 对于二分类问题,l_{ij} 属于 $\{-1,0,1\}$,其中数值分别表示众包工人对实例打负标,未给出标记和打正标. 然后通过标记集成的方法,从多噪声标记集中得到每个实例 x_i 的集成标记 $\hat{y}_i$. 因此,新的数据集 $D'=\{(x_i,\hat{y}_i)\}_{i=1}^{N}$ 是噪声纠正算法要处理的数据.

通常,噪声纠正算法包含两个步骤:过滤和纠正. 首先,通过一个噪声过滤算法识别数据集中可能是错误标记的实例,将整个数据集分为干净实例集和噪声实例集. 然后,在干净实例集上训练相应的分类模型,对噪声实例集中的实例进行重新分类,并纠正实例的标记. 其中所使用的分类模型可以是单个分类器,也可以是集成分类器.

STC 算法就如同上述过程,首先通过一个分类器过滤器将数据集分为干净实例集和噪声实例集,然后通过一个单独的分类器,比如 C4.5,对噪声实例集中的具有最高打标置信度的实例进行纠正并加入干净实例集,并重复执行这两步,直到一定比例的实例加入干净实例集. 然而,该算法仍有提升的空间,因为过滤器是在原始数据集上建立,容易受到噪声数据的影响,从而使得得到的干净实例集并不能保证足够干净. 而在纠正过程中,纠正所使用的分类器是在干净实例集上训练的,从而前一步骤的影响将损害分类器的性能,导致纠正的结果存在较大误差,而且多次循环执行该过程会积累这种误差. 那么,是否能够减小这种误差,从而提高纠正的性能? 有两种方法以削弱该误差的影响,一种通过相应的机制,在第一步提高所得到的干净实例集的干净程度;第二种则是在第二步通过集成学习等方法,

降低单个分类器造成的分类误差. 我们提出的算法 TTLNC 正是采用第二种方法,并引入半监督领域中三重训练的思想,从而提高标记噪声纠正的准确性.

2.2 TTLNC 算法

根据上节的讨论,我们希望在算法的纠正阶段采用集成学习的方法,同时借鉴三重训练的思想建立三个不同类型(异质)的分类器来降低单个分类器的误差. 同时为了进一步提高噪声纠正的准确性,我们还设计了一种新的实例分配策略,使得用于噪声纠正的分类器更加可靠. 整个算法的框架如图 1 所示.

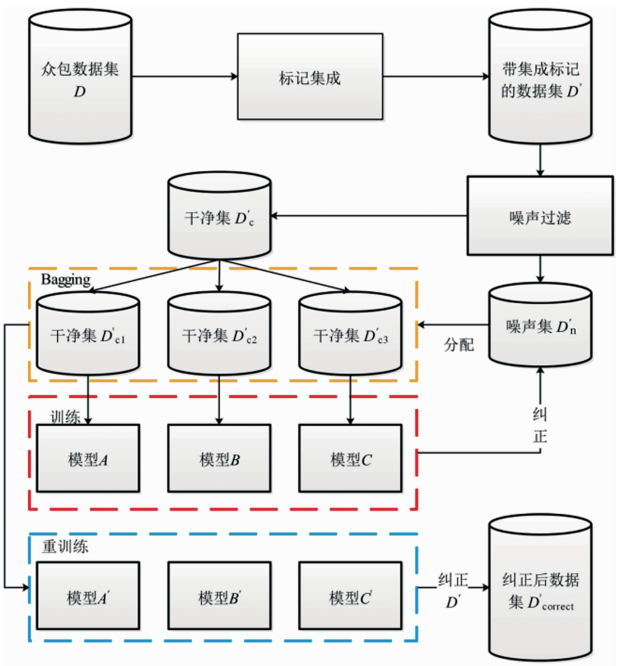

图1 TTLNC算法框架图

从图 1 中我们可以看出,首先通过标记集成算法将众包数据集 D 转化为带有集成标记的数据集 D'. 然后,通过噪声过滤器,将数据集 D' 分为干净集 D'_c 和噪声集 D'_n. 进一步将干净集 D'_c 通过 Bagging 的方法进行扰动,从而得到三个大小与 D'_c 相同而实例不同的三个数据集 D'_{c1}、D'_{c2}、D'_{c3}. 然后分别在这三个数据集上训练三个不同类型的分类器,再对噪声集 D'_n 中的每个实例进行打标,并按照实例分配策略将重新标注的实例分配给对应分类器的训练集中. 最后在新的 D'_{c1}、D'_{c2}、D'_{c3} 上重新训练这三个分类器,并用该三个分类器通过共识投票(consensus voting scheme)的策略(即:若三个分类器对某实例打标相同且不同于之前的集成标记则纠正,否则保留之前的集成标记不进行标记纠正)对 D' 中每个实例进行纠正,从而得到最终纠正后的数据集 $D'_{correct}$.

整个算法框架可以分为标记集成和标记纠正两个模块,而在标记纠正模块中,又可以分为实例过滤和实例纠正两个算法步骤. 其中,在标记集成模块中我们所使用的标记集成算法是 MV,即对于每个实例,将多噪声标记集中标记的众数,作为集成标记. 而在标记纠正模块中的实例过滤步骤,我们所使用的过滤器为 Classification Filter(CF)[14].

CF 算法是将整个数据集分为大小相等的 n 个子集,对于其中的每一个子集,都将剩余的 $n-1$ 个子集合并作为基分类器的训练集并进行训练,然后对该子集中的实例进行预测,若预测标记和集成标记不同,则该实例为噪声,并从整个数据集中去除. 上述步骤重复 n 次,直到整个数据集的实例被预测,得到最后的干净实例集. 在得到干净实例集和噪声实例集后,标记纠正模块中的实例纠正步骤的设计则是 TTLNC 算法的关键.

因此我们算法中的一个核心问题是如何设计合理的实例分配策略. 受半监督学习中三重训练思想的启发,我们提出了适合众包场景的分配策略:

(1)当三个分类器对某实例打标相同时,则将该标记赋予该实例,并将该实例分别加入到三个分类器对应的训练集中;

(2)当三个分类器对实例打标不相同时,将多数标记赋予该实例,并将该实例加入到打少数标的分类器对应的训练集中.

比如:现有分类器 A、B、C,分别对实例 x 打标,如果三个分类器都打标为 -1,则将 x 重新标注为 -1,并分别加入到三个分类器对应的训练集中;如果分类器 A 和 B 打标为 1,而分类器 C 打标为 -1,则将 x 重新标注为 1,并将该实例加入到分类器 C 对应的训练集中.

受 Tri-training 的启发,为了保证我们设计的实例分配策略的有效性,我们的算法采用了 Bagging 的方式以及选取类型差异较大的异质分类模型来保证最终三个分类器的多样性. 当三个分类器对一个实例打标相同时,那么我们则认为重新标注的该实例为干净的置信度很高,因此将该实例加入每个分类模型对应的训练集来提升最终训练模型的性能;而当其中两个分类器打标相同,另一个分类器打标不同,我们认为重新标注为多数标的实例为干净的置信度较高,因此我们将该实例加入打标不相同的分类器对应的训练集中,从而降低该分类器对该实例打标错误的可能性,提升最终训练分类器的性能. 通过对这两种情况的分析和处理,我们设计的实例分配策略能够提高分类模型的性能从而提升数据集纠正的效果.

综上所述,本文提出的 TTLNC 算法的详细步骤如算法 1 所示:

算法 1 TTLNC 算法

输入:带集成标记的数据集 $D'=\{(x_i,\hat{y}_i)\}_{i=1}^{N}$.

输出:纠正后的数据集 $D'_{correct}$.

应用 CF 过滤器过滤数据集 D',将 D'分为干净集 D'_c和噪声集 D'_n;
for $m = A$ to C do
　对干净集 D'_c进行 Bagging 得到 D'_{cm};
　在 D'_{cm}上训练分类器 m;
end for
for $i = 1$ to sizeof (D'_n) do
　用分类器 A、B、C 对噪声集 D'_n中的实例 i 进行重新标记;
　根据实例分配策略将其加入到三个分类器对应的训练数据集中;
end for
for $m = A$ to C do
　在分类器 m 的新训练集上训练得到新的分类器 m';
end for
for $i = 1$ to sizeof (D') do
　通过分类器 A'、B'、C'按照共识投票的策略对数据集 D'中的实例 i 进行纠正;
end for
得到纠正后所有实例组成的数据集 $D'_{correct}$;
return $D'_{correct}$.

3　实验设计与结果分析

在本节中,我们对提出的 TTLNC 算法在模拟工人打标的标准数据集和真实的众包数据集上进行了充分验证. 将我们提出的新算法 TTLNC 与现有的四种经典的标记噪声纠正算法 PL、STC、CC、BMNC 在纠正后的数据集噪声比以及目标模型的模型质量两个度量指标上进行了比较. 其中,数据集的噪声比(noise ratio)被定义为纠正后的数据集中集成标记与真实标记不一致的实例所占的百分比. 模型质量(model quality)定义为在纠正后的数据集上训练的目标分类模型的分类精度.

我们在 Crowd Environment and its Knowledge Analysis (CEKA)平台[15]上实现了我们提出的算法和 BMNC 算法,同时使用了该平台现有的 MV、PL、STC 和 CC 算法以及 Waikato Environment for Knowledge Analysis (WEKA)平台[16]上的 C4.5 算法,K-means 算法,K-Nearest Neighbor (KNN)算法和 Logistic Regression (LR)算法. 5 种噪声纠正算法在该实验中的具体设置如下:

(1)PL:PL 算法使用的基分类器为 C4.5;

(2)STC:CF 被用作 STC 的过滤器,而且需要被纠正的噪声实例的比例设置为 0.8. STC 算法使用的基分类器为 C4.5;

(3)CC:CC 算法中使用的聚类方法为 K-means 聚类算法,该聚类算法执行的次数设置为 10 次,而 k 的值设置为 2 到数据集实例个数的一半;

(4)BMNC:CF 被用作 BMNC 的过滤器,其中类间间隔的阈值设置为 0.2. BMNC 算法使用的基分类器为 C4.5;

(5)TTLNC:CF 被用作 TTLNC 的过滤器,其中使用的三个分类器分别为 C4.5、KNN 和 LR.

表 1　22 个模拟标准数据集的详细描述

Dataset	#Ins	#Att	#Pos	#Neg
biodeg	1055	41	356	699
breast-cancer	268	9	85	201
breast-w	699	10	241	458
credit-a	690	16	383	307
credit-g	1000	21	300	700
diabetes	768	8	268	500
heart-statlog	270	14	120	32
hepatitis	155	20	123	32
horse-colic	368	22	232	136
ionosphere	351	35	225	126
kr-vs-kp	3196	37	1527	1669
labor	57	16	37	20
mushroom	8124	23	3916	4208
sick	3772	30	231	3541
sonar	208	61	111	97
spambase	4601	57	1813	2788
tic-tac-toe	958	10	332	626
vote	435	17	168	267
climate	540	20	494	46
colic	368	22	136	232
monks	432	6	228	204
steel-plates-faults	1941	33	673	1268

注意,CF 过滤器中参数 n 表示将数据集分为同等大小的子集的个数,这里我们设置 $n = 10$,而且过滤器的基分类器同样为 C4.5.

3.1　模拟标准数据集上的实验

我们使用了与文献[12]完全相同的 22 个二分类标准数据集来完成设定的实验,数据信息具体如表 1 所示,其中"#Ins"表示数据集实例的数量,"#Att"表示数据集实例的属性维度,"#Pos"表示标记为正的实例数量,"#Neg"表示标记为负的实例数量. 为了模拟众包中为每个实例获取多噪声标记集的过程,我们隐藏了实例的真实标记,并使用了 9 个模拟工人对每个实例进行打标,其中每个工人的质量为 $p_j(j = 1,2,\cdots,9)$. 这代表每个模拟工人对实例打正确标记的概率为 p_j,而打错误标记的概率为 $1 - p_j$. 为了保证在不同工人质量下实验结果的可靠性,我们设置了两种工人质量的方案:

(1)在第一个系列的实验中,我们将所有工人的质量都设为 0.6,也就是 $p_j = 0.6$;

(2)在第二个系列的实验中,每个工人的质量从一个均匀分布的区间[0.55,0.75]中随机生成,也就是

$p_j \in [0.55, 0.75]$.

在为每个实例获得 9 个带有噪声的众包标记之后,我们使用最经典的标记集成算法 Majority Voting(MV)获得每个实例的集成标记. 然后,4 种不同的噪声纠正算法被用于识别带有集成标记数据集中的噪声实例并纠正. 接着目标分类模型将在纠正后的数据集上进行训练. 最后,我们将评估在每个数据集上各种噪声纠正算法纠正后的噪声比和训练模型质量. 值得注意的是,与评估数据集的噪声比不同,我们在评估训练模型质量的时候采用了十折交叉的验证方法,特别是在同一个数据集上运行不同算法的时候用了相同的训练集以及测试集. 在本节最后,我们还对 TTLNC 算法所选过滤器的影响进行了讨论.

表 2 和表 3 详细展示了在工人质量 $p_j = 0.6$ 的情况下各个算法在不同数据集上的实验结果. 根据表 2 中的纠正后的数据集中的噪声比以及表 3 中的模型质量,我们采用了威尔科克森符号秩检验[17,18]来比较实验用到的每一对噪声纠正算法. 威尔科克森符号秩检验是一种非参数统计测试,它对两种算法在每个数据集上表现的差异进行等级排序,然后分别根据正负等级之和判断算法间差异是否显著. 表 4 和表 5 分别展示了每组实验的威尔科克森符号秩检验的比较结果. 其中符号 · 代表该行中的算法明显优于对应列中的算法,符号 ◦ 代表该列中的算法明显优于对应行中的算法. 其中,主对角线以下区域的显著性水平为 $\alpha = 0.05$;而主对角线以上区域的显著性水平 $\alpha = 0.1$.

表 2 ~ 5 的实验结果都验证了我们提出的 TTLNC 算法在提高数据质量和模型质量两个度量指标上的有效性. 具体结论概述如下:

(1)TTLNC 算法纠正后的数据集的平均噪声比为 12.29%,明显低于 MV(27.22%)、PL(22.64%)、STC(15.88%)、CC(18.57%)和 BMNC(14.22%);

(2)TTLNC 算法纠正后的数据集训练的目标模型的平均模型质量为 84.61%,明显高于 MV(81.31%)、PL(81.48%)、STC(82.94%)和 CC(80.95%),与 BMNC(84.43%)模型质量相当;

表 2 工人质量 0.6 时噪声比(%)对比结果

Dataset	MV	PL	STC	CC	BMNC	TTLNC
biodeg	27.20	33.36	24.66	18.58	19.43	15.73
breast-cancer	24.83	27.97	26.24	30.07	27.97	23.78
breast-w	27.75	5.15	9.27	7.58	5.29	3.29
credit-a	26.52	13.91	19.81	16.52	15.65	13.33
credit-g	26.00	28.10	26.43	25.40	26.00	22.70
diabetes	27.21	27.73	22.14	23.57	23.18	21.48
heart-statlog	26.30	20.00	28.57	23.70	19.26	14.07
hepatitis	30.32	22.58	23.78	25.16	20.65	16.77
horse-colic	26.09	16.03	18.18	22.55	18.21	16.03
ionosphere	27.92	24.22	13.54	13.11	18.52	13.11
kr-vs-kp	26.10	26.28	4.69	15.33	0.91	2.88
labor	29.82	38.60	14.89	17.54	35.09	24.56
mushroom	26.65	8.16	1.52	5.34	0.11	0.06
sick	26.64	2.97	4.08	9.38	1.78	2.57
sonar	28.85	44.71	28.19	24.04	19.71	20.19
spambase	26.43	34.80	15.36	13.76	8.78	7.17
tic-tac-toe	26.93	34.66	20.81	18.58	22.55	15.03
vote	29.43	5.06	10.34	16.09	3.91	4.37
climate	27.59	8.52	3.50	13.70	8.52	8.52
colic	27.99	24.46	22.35	22.55	14.40	15.76
monks	27.08	16.20	8.62	28.47	3.01	6.71
steel-plates-faults	25.30	34.67	2.44	17.47	0.00	2.22
Average	27.22	22.64	15.88	18.57	14.22	**12.29**

表 3　工人质量 0.6 时模型质量(%)对比结果

Dataset	MV	PL	STC	CC	BMNC	TTLNC
biodeg	70.75	77.08	71.49	76.52	77.43	79.91
breast-cancer	67.96	70.34	70.34	72.06	70.34	68.55
breast-w	88.42	93.85	90.65	93.84	92.58	93.24
credit-a	84.35	85.65	85.22	84.06	85.36	85.07
credit-g	67.30	70.20	70.70	68.10	69.50	72.00
diabetes	68.08	69.38	66.89	70.41	69.86	71.18
heart-statlog	72.96	73.70	73.33	72.59	78.89	75.19
hepatitis	64.17	75.83	79.17	81.67	85.00	80.67
horse-colic	83.51	66.04	83.56	84.67	84.12	85.78
ionosphere	79.21	79.48	82.06	84.02	83.48	82.05
kr-vs-kp	95.30	94.84	96.56	89.99	96.71	97.46
labor	72.33	70.67	74.67	69.00	71.50	77.50
mushroom	99.57	98.52	99.61	99.73	99.67	99.82
sick	96.74	96.29	97.69	95.09	98.12	97.24
sonar	54.50	60.00	56.00	62.71	65.93	62.21
spambase	87.05	89.70	87.65	85.76	90.63	90.33
tic-tac-toe	74.11	71.15	73.88	74.30	74.58	78.43
vote	93.54	95.63	94.23	91.96	93.54	94.47
climate	91.48	91.48	91.48	91.48	91.48	91.48
colic	83.84	80.78	83.84	80.28	83.11	83.51
monks	93.73	86.54	95.59	84.03	95.59	95.36
steel-plates-faults	99.95	95.31	100.00	93.56	100.00	100.00
Average	81.31	81.48	82.94	82.08	84.43	**84.61**

表 4　工人质量 0.6 时噪声比的威尔科克森测试

	MV	PL	STC	CC	BMNC	TTLNC
MV	–		∘	∘	∘	∘
PL		–	∘		∘	∘
STC	·	·	–	∘	∘	∘
CC	·			–	∘	∘
BMNC	·	·	·	·	–	∘
TTLNC	·	·	·	·	·	–

表 5　工人质量 0.6 时模型质量的威尔科克森测试

	MV	PL	STC	CC	BMNC	TTLNC
MV	–		∘		∘	∘
PL		–			∘	∘
STC	·		–		∘	∘
CC				–	∘	∘
BMNC	·	·	·	·	–	
TTLNC	·	·	·	·		–

(3)根据威尔科克森符号秩检验的结果,TTLNC 算法在噪声比方面要显著优于其他噪声纠正算法. 而在模型质量方面要显著优于 MV、PL、STC 和 CC,与 BMNC 性能相当.

在第二个系列的实验中,表 6 ~ 9 详细展示了在工人质量 $p_j \in [0.55, 0.75]$ 的情况下各个噪声纠正算法纠正后数据集的噪声比和模型质量的实验结果,以及在威尔克森符号秩检验中的算法差异显著性的对比结果. 根据表 6 和表 7 中的实验结果,基本的实验结论与第一个系列的实验相似,在数据集的噪声比和模型质量两个度量指标上,TTLNC 算法的效果都是最好的. 同时,根据表 8 和表 9 中的威尔克森符号秩检验的结果,可以看出 TTLNC 算法在第二个系列的实验中表现更好,在纠正后数据集的噪声比上显著优于对比算法 MV、PL、CC 和 BMNC,与 STC 算法相当. 而在模型质量上显著优于对比算法 MV、PL、STC、CC 和 BMNC.

根据以上两个系列的实验结果,我们在不同工人质量的模拟数据集上都验证了我们提出的 TTLNC 算法在提高数据质量和模型质量上的有效性,并且 TTLNC 算法在纠正后数据集的噪声比和模型质量两个度量指标上整体优于对比的四个标记噪声纠正算法.

为了进一步验证 TTLNC 算法的有效性,这里针对所选过滤器对 TTLNC 的影响进行了讨论. 在设计的实验中,根据数据集样本和维度大小以及正负样本比例,我们选择了三个差异较大且具有代表性的数据集 breast-cancer、credit-g 和 kr-vs-kp,详细描述如表 1 所示.

同时,我们选择了传统机器学习领域经典的噪声过滤器 Edited Nearest Neighbor rule(ENN)[19]、Majority Vote Filter(MVF)[20]和 Classification Filter(CF),三个过滤器具体描述如下:

表 6 工人质量在[0.55,0.75]时噪声比(%)对比结果

Dataset	MV	PL	STC	CC	BMNC	TTLNC
biodeg	23.51	25.97	18.03	17.35	16.30	12.61
breast-cancer	14.34	27.27	22.01	22.03	23.78	23.78
breast-w	15.74	4.01	6.09	3.86	3.72	2.72
credit-a	16.23	16.23	12.29	13.48	12.61	11.59
credit-g	18.60	25.40	21.97	21.70	25.10	22.60
diabetes	18.88	23.05	18.58	22.53	21.09	21.48
heart-statlog	20.37	16.30	18.55	20.74	19.63	17.78
hepatitis	14.84	13.55	15.86	12.26	19.35	18.06
horse-colic	18.48	14.13	12.06	16.85	13.32	13.59
ionosphere	16.24	15.95	11.68	9.40	8.83	8.83
kr-vs-kp	17.58	24.03	2.83	12.86	1.13	1.60
labor	22.81	38.60	16.00	19.30	21.05	14.04
mushroom	15.02	7.26	0.52	0.60	0.00	0.00
sick	14.74	1.75	2.16	3.71	1.25	1.56
sonar	20.67	34.62	19.89	18.27	20.67	14.90
spambase	15.28	33.69	9.47	8.04	6.59	5.02
tic-tac-toe	19.62	34.66	17.46	16.28	18.27	10.65
vote	16.09	3.91	6.00	8.74	4.37	4.60
climate	15.56	8.52	2.55	9.26	8.52	8.52
colic	19.29	14.13	18.39	17.39	14.13	12.23
monks	20.83	21.99	1.95	22.45	3.01	0.46
steel-plates-faults	11.59	34.67	1.42	11.28	0.00	0.77
Average	17.56	19.99	11.63	14.02	11.94	**10.34**

表 7 工人质量在[0.55,0.75]时模型质量(%)对比结果

Dataset	MV	PL	STC	CC	BMNC	TTLNC
biodeg	79.71	82.65	80.36	81.14	81.42	83.79
breast-cancer	70.34	70.99	70.57	73.84	70.27	69.98
breast-w	93.56	94.88	93.90	94.14	95.14	94.74
credit-a	85.65	85.22	84.35	85.07	84.93	84.35
credit-g	71.80	69.60	72.70	70.10	71.50	73.70
diabetes	70.78	69.09	72.44	73.42	76.02	74.86
heart-statlog	75.56	75.19	72.22	75.93	78.15	77.41
hepatitis	80.00	79.67	79.17	80.83	82.33	80.67
horse-colic	84.17	84.72	86.11	81.94	85.88	85.83
ionosphere	85.20	89.47	85.18	85.47	87.77	84.33
kr-vs-kp	99.09	94.46	99.06	94.18	99.09	98.97
labor	74.00	72.67	73.17	76.67	77.17	79.17
mushroom	100.00	98.52	100.00	100.00	99.98	100.00
sick	98.20	97.67	98.06	96.16	98.06	98.01
sonar	67.57	72.29	65.07	62.79	67.21	71.29
spambase	88.96	90.59	89.59	88.16	90.98	91.65
tic-tac-toe	79.05	70.80	74.69	73.83	76.31	78.58
vote	94.75	95.63	94.96	93.38	95.40	95.19
climate	91.48	91.48	91.48	91.48	91.48	91.48
colic	82.25	80.86	83.69	81.41	83.41	81.19
monks	94.66	96.28	96.74	88.61	97.21	96.74
steel-plates-faults	100.00	99.64	100.00	98.25	100.00	100.00
Average	84.85	84.65	84.71	83.95	85.90	**86.00**

表 8　工人质量在[0.55,0.75]时噪声比的威尔科克森测试

	MV	PL	STC	CC	BMNC	TTLNC
MV	–		∘	∘	∘	∘
PL		–	∘	∘	∘	∘
STC	·	·	–			∘
CC	·	·		–		∘
BMNC	·	·		·	–	∘
TTLNC	·	·		·	·	–

表 9　工人质量在[0.55,0.75]时模型质量的威尔科克森测试

	MV	PL	STC	CC	BMNC	TTLNC
MV	–		∘	∘	∘	∘
PL		–	∘	∘	∘	∘
STC	·	·	–			∘
CC	·	·		–		∘
BMNC	·	·			–	∘
TTLNC	·	·	·	·	·	–

(1)ENN:ENN 的思想是基于 KNN 算法,对数据集中每个实例进行分类,若得到的标记与原标记不同,则该实例为噪声,并从数据集中去除. 因此 ENN 所使用的基分类器为 KNN,且 k 值设置为 3;

(2)MVF:MVF 的思想是将整个数据集分为大小相等的 n 个子集,对于其中的每一个子集,都将剩余的 $n-1$ 个子集合并作为 m 个基分类器的训练集并进行训练,然后对该子集中的实例进行预测,再根据多数投票的策略判断该实例是否为噪声. 上述步骤重复 n 次,直到整个数据集的实例被预测. 在本实验中 MVF 的 n 值设置为 10,m 值设置为 3,所使用的基分类器为 KNN、Naïve Bayes 和 C4.5,其中 KNN 的 k 值设置为 1;

(3)CF: CF 的思想是将整个数据集分为大小相等的 n 个子集,对于其中的每一个子集,都将剩余的 $n-1$ 个子集合并作为基分类器的训练集并进行训练,然后对该子集中的实例进行预测,然后对该子集中的实例进行预测,若预测标记和集成标记不同,则该实例为噪声,并从整个数据集中去除. 上述步骤重复 n 次,直到整个数据集的实例被预测. 在本实验中 CF 的 n 值设置为 10,基分类器使用 C4.5.

根据以上的实验设置,我们分别在工人质量为 0.5、0.6、0.7、0.8 以及 0.9 的情况下进行了实验,并评估了使用不同过滤器的 TTLNC 在数据集纠正后的噪声比和训练模型质量.

图 2 ~7 分别展示了在不同工人质量下使用各个过滤器的 TTLNC 算法在三个数据集上的噪声比和模型质量的详细对比结果. 从对比结果可以看出:在不同工人质量以及数据集上使用 ENN、MVF 和 CF 过滤器的 TTLNC 在噪声比和模型质量上结果相似. 因此 TTLNC 算法在选用不同过滤器时表现稳定,对算法性能影响不大. 为了提高算法效率,且与 STC 算法使用的过滤器保持一致,在本文中 TTLNC 所使用的为 CF 过滤器.

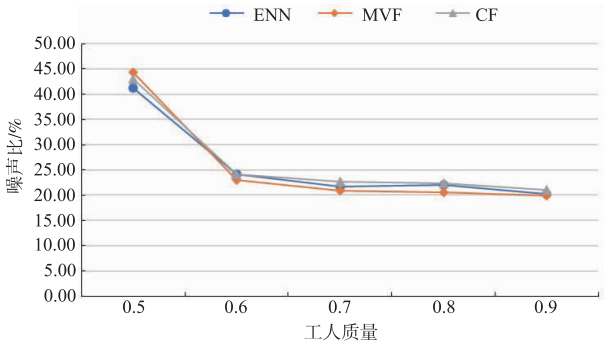

图2　噪声比(%)在breast-cancer数据集上的对比结果

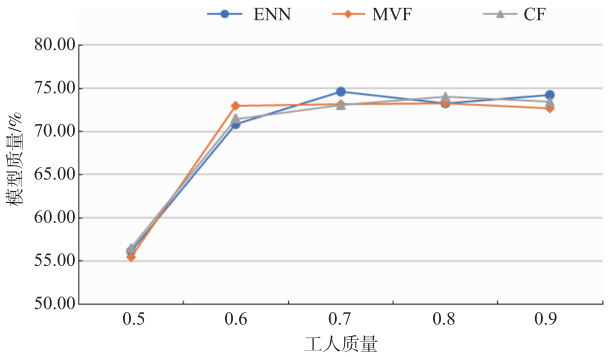

图3　模型质量(%)在breast-cancer数据集上的对比结果

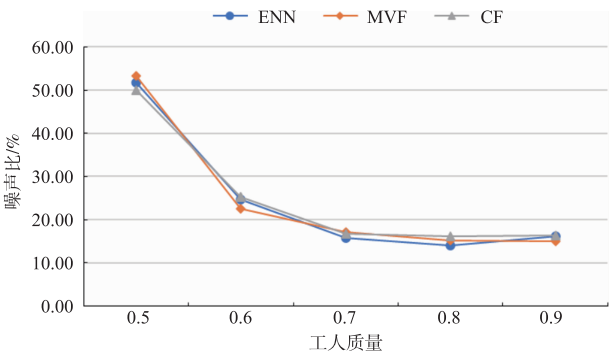

图4　噪声比(%)在credit-g数据集上的对比结果

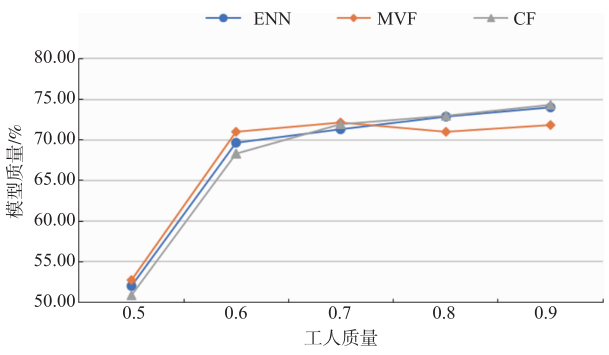

图5　模型质量(%)在credit-g数据集上的对比结果

3.2　真实众包数据集上的实验

为进一步验证 TTLNC 算法的有效性,本节使用

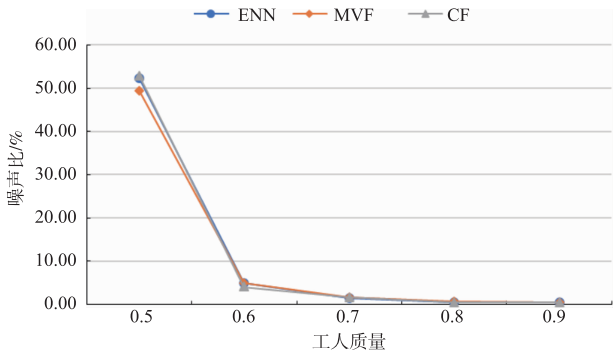

图6 噪声比(%)在kr-vs-kp数据集上的对比结果

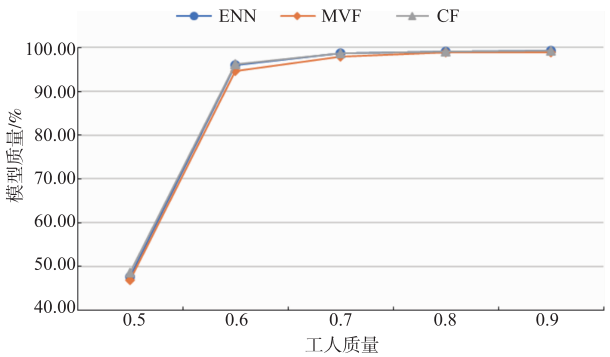

图7 模型质量(%)在kr-vs-kp数据集上的对比结果

CEKA 平台自带的真实众包数据集 Leaves 进行实验. 该数据集包含 384 个带有 63 维特征的实例. 每个实例都是一张树叶的图片,这些图片被发布在 AMT 平台上,由众包工人根据树叶的颜色和形状判断树叶的种类. 因此,Leaves 数据集中每个实例都获得多个众包工人的标注.

为了适用于本文的二分类算法,我们从Leaves中提取了四个只包含二类的数据集. 这四个数据集分别被记为 Leaves1、Leaves2、Leaves3 和 Leaves4,这些数据集的具体细节如表 10 所示. 例如:数据集“Leaves1”的任务是判断实例为枫树叶还是桤木树叶的图片. 数据集中包含 142 个实例,其中 46 个为正例(枫树叶),96 个为负例(桤树叶). 为了获得每个实例的多噪声标记集,总共有 70 个众包工人标注了 1093 个标记.

表 10 四个真实众包数据的详细描述

Dataset	Task	#Instances	#Positives	#Negatives	#Labelers	#Labels
Leaves1	maple/alder	142	46	96	70	1093
Leaves2	alder/poplar	89	43	46	53	400
Leaves3	eucalyptus/oak	140	45	95	66	930
Leaves4	poplar/oak	143	48	95	68	994

图 8 和图 9 分别展示了各个算法在四个真实众包数据集上纠正后的噪声比和模型质量的详细对比结果. 从对比结果可以看出:我们提出的 TTLNC 算法整体优于 MV、PL、STC、CC 以及 BMNC 算法,可以得出在模拟数据集上几乎相同的结论. 因此,通过上述实验的比较,验证了 TTLNC 算法在真实众包场景的有效性,同时提高了数据质量和模型质量.

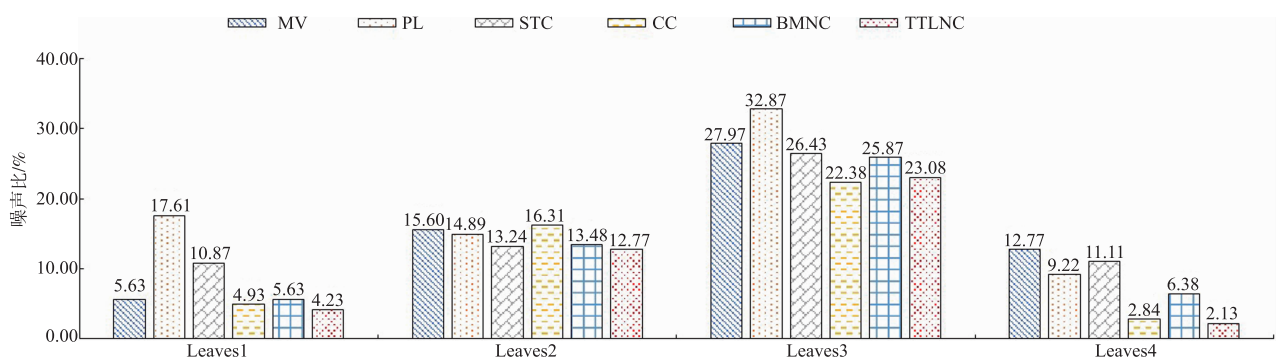

图8 噪声比(%)在4个真实众包数据集上的对比结果

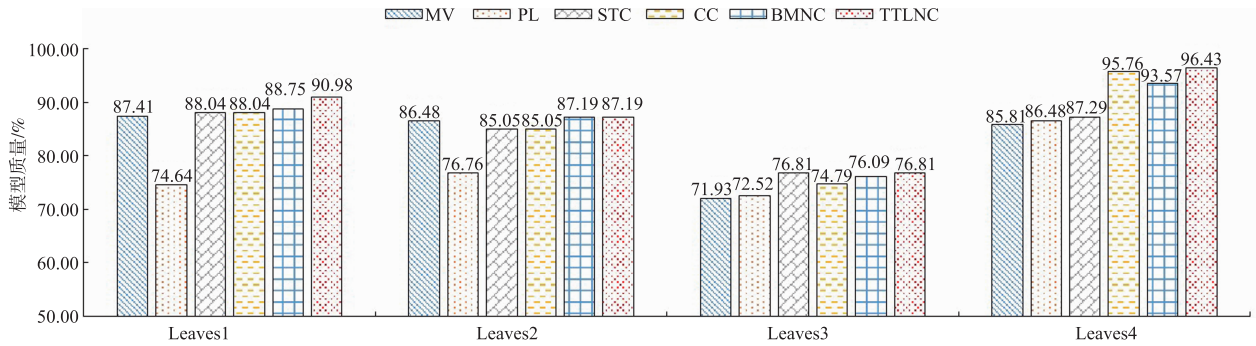

图9　模型质量(%)在4个真实众包数据集上的对比结果

4　总结

本文提出了一种基于 Tri-training 的众包标记噪声纠正算法 TTLNC. 通过提出的实例分配策略,同时引进 Tri-training 和集成学习的思想,从而进一步提高了分类模型的可靠性,最终实现了对众包标记噪声数据更准确地识别与纠正. 根据在 22 个模拟的标准数据集以及 4 个真实的众包数据集上的实验结果,TTLNC 与 PL、STC、CC、BMNC 四种目前最先进的噪声纠正算法相比,其性能在数据集噪声比和模型质量两个度量指标上更好,从而验证了所提出的标记噪声纠正算法的有效性和优越性. 但是,目前的版本还没有考虑多分类情况下实例分配策略的设计,不能应用于多分类问题中. 因此未来的工作将尝试对实例分配策略进行更精细化的设计,以进一步拓展新算法在多分类问题中的应用.

参考文献

[1] Zhang H, Jiang L, Xu W. Differential evolution-based weighted majority voting for crowdsourcing[A]. Proceedings of the 15th Pacific Rim International Conference on Artificial Intelligence[C]. Berlin, Heidelberg: Springer, 2018. 228 – 236.

[2] Sheng V S, Provost F, Ipeirotis P G. Get another label? improving data quality and data mining using multiple, noisy labelers[A]. Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. New York, USA: ACM, 2008. 614 – 622.

[3] Demartini G, Difallah D E, CudreMauroux P. ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking[A]. Proceedings of the 21th International Conference on World Wide Web[C]. New York, USA: ACM, 2012. 469 – 478.

[4] Ma F, Li Y, Li Q, et al. FaitCrowd: fine grained truth discovery for crowdsourced data aggregation[A]. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. New York, USA: ACM, 2015. 745 – 754.

[5] Zhang J, Sheng V S, Wu J, et al. Multi-class ground truth inference in crowdsourcing with clustering[J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 28(4): 1080 – 1085.

[6] Tian T, Zhu J. Max-margin majority voting for learning from crowds[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2018, 41(10): 2480 – 2494.

[7] Zhang J, Sheng V S, Li T. Label aggregation for crowdsourcing with bi-layer clustering[A]. Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval[C]. New York, USA: ACM, 2017. 921 – 924.

[8] Zhang H, Jiang L, Xu W. Multiple noisy label distribution propagation for crowdsourcing[A]. Proceedings of the 28th International Joint Conference on Artificial Intelligence[C]. Palo Alto, USA: AAAI Press, 2019. 1473 – 1479.

[9] Sheng V S, Zhang J, et al. Majority voting and pairing with multiple noisy labeling[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(7): 1355 – 1368.

[10] Nicholson B, Sheng V S, Zhang J. Label noise correction and application in crowdsourcing[J]. Expert Systems with Applications, 2016, 66: 149 – 162.

[11] Zhang J, Sheng V S, Li T, Wu X. Improving crowdsourced label quality using noise correction[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(5): 1675 – 1688.

[12] Li C, Jiang L, Xu W. Noise correction to improve data and model quality for crowdsourcing[J]. Engineering Applications of Artificial Intelligence, 2019, 82: 184 – 191.

[13] Zhou Z, Li M. Tri-training: exploiting unlabeled data using three classifiers[J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(11): 1529 – 1541.

[14] Gamberger D, Lavrac N, Groselj C. Experiments with noise filtering in a medical domain[A]. Proceedings of the 16th International Conference on Machine Learning[C]. Amsterdam, the Netherlands: Elsevier, 1999. 143 – 151.

[15] Zhang J, Sheng V S, Nicholson B, et al. CEKA: a tool for mining the wisdom of crowds[J]. Journal of Machine Learning Research, 2015, 16(1): 2853 – 2858.
[16] Witten I H, Frank E. Data Mining: Practical Machine Learning Tools and Techniques (Third Edition)[M]. Beijing: China Machine Press, 2005.
[17] Garcia S, Herrera F. An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons[J]. Journal of Machine Learning Research, 2008, 9(12): 2677 – 2694.
[18] Jiang L, Zhang L, Li C, Wu J. A correlation-based feature weighting filter for naive Bayes[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(2): 201 – 213.
[19] Wilson, Dennis L. Asymptotic properties of nearest neighbor rules using edited data[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1972, 3: 408 – 421.
[20] Brodley C E, Friedl M A. Identifying mislabeled training data[J]. Journal of Artificial Intelligence Research, 1999, 11: 131 – 167.

作者简介

杨 艺 男,1996 年 5 月出生于江西省九江市. 现为中国地质大学(武汉)计算机学院研究生. 主要研究方向为机器学习和数据挖掘.
E-mail: yangyi@cug.edu.cn

蒋良孝(通信作者) 男,1977 年 4 月出生于湖南省衡阳市. 现为中国地质大学(武汉)教授、博士生导师. 主要研究方向为机器学习和数据挖掘.
E-mail: ljiang@cug.edu.cn

李超群 女,1981 年 2 月出生于湖北省松滋市. 现为中国地质大学(武汉)副教授、硕士生导师. 主要研究方向为机器学习和数据挖掘.
E-mail: chqli@cug.edu.cn

李宏伟 男,1964 年 4 月出生于湖南省汨罗市. 现为中国地质大学(武汉)教授、博士生导师. 主要研究方向为智能计算与信息处理.
E-mail: hwli@cug.edu.cn