一种基于预过滤和聚类处理的众包标签噪声纠正方法

史 伟 李超群*
(中国地质大学(武汉)数学与物理学院 湖北 武汉 430074)

摘 要 面向众包标注数据,提出一个新的标签噪声纠正方法 MCNC(modified cluster-based noise correction)。利用实例多标签集合的信息进行预过滤,构建过滤器进行二次噪声过滤。在原始数据集上进行聚类学习,对两次过滤中去除的实例进行重新标注。在 22 个数据集上的实验结果表明,MCNC 可以有效提升数据集的集成标签质量,从而提高目标分类器的性能。

关键词 众包学习 集成标签 标签噪声 噪声纠正

中图分类号 TP181 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2022.07.002

A CROWDSOURCING LABEL NOISE CORRECTION METHOD BASED ON PRE-FILTERING AND CLUSTERING

Shi Wei Li Chaoqun*
(*School of Mathematics and Physics, China University of Geosciences, Wuhan 430074, Hubei, China*)

Abstract This paper proposed a new label noise correction method, which named modified cluster-based noise correction(MCNC), for crowdsourcing labeling data. The MCNC used the information of the instance multiple label set to prefilter, and then built a filter to filter the secondary noise. Cluster learning was carried out on the data set, and the instances removed from the two filters were re-labeled. The experimental results on 22 data sets show that MCNC can effectively improve the quality of the integrated label of the data set, thus improving the performance of the target classifier.

Keywords Crowdsourcing learning Integrated labels Label noise Noise correction

0 引 言

带标签的训练数据对监督学习是必不可少的,然而从领域专家处获得数据的标签往往昂贵且耗时[1]。在众包技术的帮助下,可以以较低的成本雇佣众包工人(标注者)对数据集中的实例进行标注。但由于种种原因,例如,工人的报酬低、工人欠缺专业知识、实例本身标注的困难性等,单个工人的标注质量可能比较低。为此,一个常见的解决方法是让多个不同的工人对同一个实例进行标注(即重复标注),然后用标签真值推理算法去推理得到每个实例的集成标签。详细而言,$U=\{u_j\}_{j=1}^{R}$表示众包系统中的 R 位工人,每个实例 x_i有一个多标签集合 $I_i=\{l_{ij}\}_{j=1}^{R}$,其中 l_{ij}指实例 x_i从工人 u_j处获得的标签。以一个有 5 个工人的二分类众包系统为例,实例 x 的多标签集合为 $I=\{+,-,+,+,-\}$。在每个实例获得其多标签集后,使用真值推理算法可以得到实例的集成标签,记为$\hat{y}_i$。这时,就得到了可用于训练模型的数据集 $\hat{D}=\{(x_i,\hat{y}_i)\}_{i=1}^{N}$。

为了获得实例的集成标签,最简单的方法就是 Ipeirotis 等[2]提出的众数投票法(Majority Voting,MV),即将实例 x_i的多标签集的多数类赋予该实例。但是 MV 比较粗糙,为了获得更高质量的集成标签,很多研究者致力于设计新的标签真值推理算法。例如 Raykar 等[3]提出 RY,Demartini 等[4]提出 ZC,Sheng[5]提出 MV 的两种变形 MV-Freq 和 MV-Beta,Tian 等[6]提出的 M3V,Zhang 等[7]提出的 DEWMV。

毫无疑问,无论使用哪一种真值推理算法,集成标签中仍存在一定程度的噪声。这里噪声是指与专家标签不同的集成标签,集成标签为噪声的实例本文称为

收稿日期:2020-03-29。国家级大学生创新创业训练项目(G1320311901)。**史伟**,学士,主研领域:数据挖掘与机器学习。**李超群**,副教授。

噪声实例。在监督学习中,训练数据集的标签质量对于模型的构建至关重要。因为错误标注的数据可能会降低目标分类器的性能,同时增加模型训练的复杂度,所以噪声处理是十分有必要的。在信号处理领域,噪声处理是一个相对成熟的问题。虽然大多数工作可以直接用于集成标签进行噪声过滤或纠正,但这些工作不是针对众包数据设计的,不能有效利用众包系统所产生的信息。现有文献中,将众包噪声处理与机器学习结合的研究并不多。另外,虽然很多噪声过滤方法可以有效地过滤集成标签中的噪声数据,但是简单抛弃一部分实例无疑是一种数据浪费,而标签纠正技术可以减少这一浪费。基于以上两点,本文致力于众包数据的标签噪声纠正技术,利用实例多标签集中的信息,提出一个新的标签噪声纠正方法,称为修改的基于聚类的噪声纠正算法(MCNC)。多标签集中的信息用于监督标签噪声纠正过程。在若干数据集上的实验结果表明,对比其他标签噪声纠正方法,该方法可以更有效地提高数据质量和目标模型质量。

1　相关工作

面向众包机器学习,研究者们展开了很多工作。其中主要的工作集中在真值推理算法的改进。还有的工作是将众包学习和主动学习结合,这个方向的核心观点是如何选择最不确定的实例[8-9]进行标注;也有一部分工作是将众包生成的多标签集和专家标签结合[10-12],通过引入小部分专家标签(黄金数据)来提升集成标签的质量,本质上仍属于标签真值推理的范畴。尽管在提升集成标签质量方面研究者们做了很大的努力,一个在若干个实际众包数据集上的实证研究表明这些算法的表现没有显著差异[13]。为了进一步提升集成标签的质量,进行标签噪声处理是十分有必要的。

一份关于标签噪声的综述[14]回顾了标签噪声鲁棒、标签噪声清理和标签噪声容忍模型。而标签噪声清理往往比建立噪声容忍学习模型[15-17]更加有效。存在很多可行的噪声清理方法,例如,基于度量和阈值的方法[18]通过一种特殊的度量评价了每一个实例,如果某个实例在某方面的度量超过预定阈值,则这个实例将被视为噪声被去除。K 近邻(KNN)的方法使用了 KNN 分类器对标签噪声敏感的特点;基于 KNN 的方法移除被其他实例认为是噪声的实例。基于模型影响和内省的方法[19-20],通过分析错误标记实例对模型的影响来检测错误标记实例。一种更通用的噪声过滤的方法是基于模型预测的过滤,其通过训练一个学习模型去分类实例并发现噪声,例如,classification filtering(CF)、voting filtering(VF)和 partition filtering(PF)[21]。实际上,很少有研究专门针对众包领域的噪声处理。Li 等[22]验证了通过噪声过滤技术可以提高众包数据的集成标签质量和目标分类器的性能。

相比噪声过滤,标签噪声纠正的算法要少一些。尽管噪声过滤器是有效的处理噪声的方法,但可能会过滤过多实例[14],使得剩余的实例不足以训练一个好的分类器,并且对于数据资源是一种极大的浪费。因此,本文致力于通过噪声纠正技术提升众包学习的性能。下面将回顾现有文献中的噪声纠正技术。

Nicholson 等[23]提到了三种标签噪声纠正方法:Polishing Labels(PL)、Self-Training Correction(STC)和 Cluster-based Correction(CC)。PL 是由 Teng[24]中 Polishing 方法改进而来,将其从关注属性噪声的纠正更改为关注标签噪声的纠正。该方法首先将数据集分成十个部分,然后用单一的分类算法在每一个部分上建立一个模型,用十个模型对数据集中每一个实例进行分类,得票多的标签被赋予这个实例。STC 灵感来源于 Triguero 等[25]的自训练,具体而言,STC 首先在数据集上使用一个噪声过滤器生成一个干净数据集和噪声数据集,然后在干净数据集上训练一个模型用于计算噪声数据集中每个实例是某个标签的置信度,置信度最高的标签被赋予这个实例,并将这个实例加入到干净数据集中。重复这一过程,直到一定比例的噪声实例被重新标记并加入干净数据集。不同于上述两种方法,CC 是基于聚类的方法,该方法能形成独立的簇标签,无视数据集中的噪声等级。CC 的基本思想是在数据集上执行多次聚类算法,根据每个簇中实例标签的分布和簇的大小,为每个簇中的所有实例赋予相同的权重,权重反映了实例属于不同标签的可能性。最终每个实例对从不同簇中得到的权重求和,并将对应最大权重的标签赋予该实例。CC 因为多次聚类的原因,有较高的时间复杂度,但作者的实验结果表明,相比于 PL 和 STC,CC 的性能更好。

但上面提到的方法都不是为众包学习特别设计的。据我们所知,只有两个噪声纠正方法是特别为众包学习设计的。一个是自适应投票噪声纠正方法(Adaptive Voting Noise Correction, AVNC)[26]。AVNC 通过真值推理阶段得到的信息监督噪声识别过程,同时在众包系统中使用工人的标注质量去估计数据集中噪声的数量。不仅如此,AVNC 还对噪声实例进行排序,以此来决定哪些实例更应该被去除。然后 AVNC

利用集成学习模型来纠正噪声实例的标签。AVNC 的优势是使用了真值推理阶段的信息(即工人的标注质量)来监督噪声的识别和过滤。但是 AVNC 仅仅关注了数据质量而没有关注模型质量。另一个是基于类别间隔的噪声纠正方法(Between-class Margin-based Noise Correction,BMNC)[27]。文章认为如果用于构建过滤器的数据集本身是带有噪声的,那么过滤器将不可避免地被噪声实例影响,导致产生的干净数据集并不完全干净。所以 BMNC 在进行噪声过滤之前,利用真值推理阶段的信息进行一次预过滤,去除一些潜在的噪声实例;然后训练一个分类器用于进一步分离出噪声实例,这是第二步过滤。经过两步过滤后得到一个干净数据集和一个噪声数据集,在干净数据集上构建分类器,用于对噪声数据集中的实例进行重新标注。

本文致力于结合众包系统的信息和聚类算法,设计一个新的标签噪声纠正方法,本文算法同时关注标签质量和模型质量。

2 算法设计

一般而言,一个标签噪声纠正方法包括两个步骤:噪声识别和噪声纠正。噪声识别最常见的一类方法是在数据集上建立分类器,利用分类器的预测标签与实例本身的标签进行对照,从而识别哪些实例是噪声实例。在识别出噪声标签后,再进行校正。但由于数据标签本身带有噪声,直接在这样的数据上建立分类器进行噪声识别,势必会限制噪声识别的性能,导致噪声识别准确率不理想。面向众包数据,BMNC 算法[27]使用众包数据的多标签集信息对噪声进行了一次预过滤,去除部分潜在噪声是有效且必要的。

BMNC 算法通过使用每个实例的多标签集合中的信息来对数据集进行预过滤。通过让不同的工人对同一个实例进行标注,每个实例 x_i 会有一个多标签集 I_i。用 N_l 表示多标签集 I_i 中标签 l 的数量,p_l 表示标签 l 出现的比例,则:

$$p_l = \frac{N_l}{\sum_{l \in L} N_l}$$

式中:L 是数据集的标签集合,包含了数据集中所有可能的标签取值。

在实例的多标签集合中,若各个标签获得工人投票的差距不大,则认为这个实例的集成标签有更大的可能成为噪声。当众包系统中工人没有足够的专业知识或者该实例本身较难以标注时会发生这样的现象。BMNC 中使用 $|p(+)-p(-)|$ 度量每个实例是噪声的可能性,其中 $p(+)$ 和 $p(-)$ 分别是实例的多标签集中正标签和负标签的比例。$|p(+)-p(-)|$ 越小,说明该实例获得的正负标签数目越接近,因此该实例的集成标签更有可能是噪声。但是 $|p(+)-p(-)|$ 仅能计算二分类问题,扩展到多分类问题,熵是一个比较好的解决方法。熵的定义为 $Entropy = -\sum_{l \in L} p_l \log(p_l)$。本文将关注多类分类问题,不仅是二分类问题。因此本文采用熵作为度量来对噪声进行预过滤。当实例的多标签集合中,不同标签的比例较为接近(此时工人对该实例的标注意见分歧越大),该实例的熵会越大,则该实例是噪声的可能性更大。结合二分类问题考虑,当 $p(+)$ 和 $p(-)$ 分别是 0.6 和 0.4,$|p(+)-p(-)| \leqslant 0.2$ 时,我们认为正负标签的比例接近,实例有较大可能是噪声。对应地,取对数函数底数为 2 时,此时熵值近似取为 0.95。因此,本文实验中将阈值设为 0.95,即对某个实例,若其熵值大于 0.95,该实例的集成标签被认为是噪声,将该实例过滤。

之后,在预过滤后的数据集上再过滤,用于进一步识别出噪声。具体的做法是在预过滤的数据集上建立分类器,利用分类器对预过滤后的数据集进行分类,若一个实例所获得的分类器的预测标签不同于该实例的集成标签,则该实例被判定为噪声。经过两步过滤后,已经识别出所有可能的噪声实例。

对噪声实例的纠正,比较普遍的做法之一也是在数据集上建立分类器,用分类器对噪声实例进行预测,将预测的标签赋予噪声实例,达到对噪声实例的标签进行纠正的目的,比如 STC 方法。但本文拟采用聚类方法 CC,在原始数据集上进行多次聚类,利用 CC 的思想对噪声实例赋予新的标签。之前的工作已经表明,相较于 STC,CC 的效果更好。可能的原因在于:(1) CC 的方法是基于聚类的方法,因此方法的性能本身与数据的标签质量无关;(2) CC 的方法进行了多次聚类,形成了许多簇,因此是一个类似集成学习的思路。关于监督学习的研究已经表明,基于集成学习的分类器往往比单分类器要显示出更好的分类性能。CC 可以看成是基于聚类的集成学习,利用多次聚类的思路,既克服了单次 k 均值算法对 k 值大小敏感的问题,也在多次聚类中利用大小不同的簇对实例的可能类标赋予不同的权值,通过权值求和得到实例的最终标签。

基于上述讨论,本文在构建过滤器去识别噪声之前,对数据集进行一个预过滤。通过预过滤,去除一些潜在的噪声实例。之后,在预过滤后的数据集上训练

一个分类器,用于进一步识别出噪声实例。经过两步过滤后,已经识别出所有可能的噪声实例。接下来是对噪声实例的纠正,使用原始数据集中所有的实例进行多次聚类,利用聚类结果对前两步识别出的噪声实例进行重新标注。将本文算法称为修改的基于聚类的噪声纠正算法(MCNC)。

MCNC 方法细节如算法 1 和算法 2 所示。算法 1 中,1 -7 行使用熵对数据集中的实例进行初步过滤。8 -9 行进行了第二次过滤。为了解决数据集中标签不均衡问题,10 -12 行计算了数据集中指定标签的分布,该信息被用于算法 2 中,计算每个实例的聚类标签权重。13 -21 行给出了所有需要的聚类算法,这里使用 k 均值聚类,k 取值从 2 到集合中实例数的一半不等。这个过程会产生大量不同大小的簇,用于增加聚类得到的簇的多样性。在每次聚类结果中,按照簇的不同,依据簇中所有实例的标签分布,计算该簇整体是各个类标的可能性,即算法 2 计算的权值。18 行对簇中每个实例进行一个权值的累加,即每个实例是各标签的可能性。22 -25 行使用聚类产生的标签权重对噪声实例进行重新标注。算法 2 对算法 1 中的 18 行的 CalcWeights 进行了详细解释,说明了如何根据数据集中标签分布和具体簇中的标签分布计算各标签权重。第 1 行计算了具体簇中的标签分布,第 2 行计算各标签的预期分布,第 3 行是一个乘数,用于给较大的簇更大的重要级,但包含 100 个实例以上的簇获得最大的重要级是 2,是为了不让非常大的簇淹没较小的簇。4 -6 行计算了该簇是各个标签的权重,簇中标签的实际分布减去标签的预期分布,并按照数据集中的标签分布进行缩放,乘以乘数得到权重。

算法 1　MCNC 流程

输入:一个带有集成标签的数据集 $\hat{D}=\{(x_i,\hat{y}_i)\}_{i=1}^{N}$,数据集 $\hat{D}$ 的多标签集合 $\{I_i\}_{i=1}^{N}$,阈值 δ,聚类次数 a,标签集合 L。

输出:纠正后的数据集 $\tilde{D}$。

1. 一个空的集合 A;
2. **for** $i=1$ to N **do**
3. 计算多标签集合 I_i 中每个标签的比例 p_l;
4. **if** $-\sum_{l\in L}p_l\log(p_l)\geqslant\delta$
5. 将实例 x_i 添加到集合 A 中;
6. **end if**
7. **end for**
8. 在集合 $\tilde{D}\setminus A$ 上应用一个过滤器,并将过滤出的实例记为 B;
9. $\tilde{D}=\tilde{D}\setminus(A+B)$;
10. **for** $i=1$ to N **do**
11. $LableTotals[\hat{y}_i]=LableTotals[\hat{y}_i]+1$;
12. **end for**
13. **for** $i=1$ to a **do**
14. $k=\frac{i}{a}\times\frac{N}{2}+2$;
15. $C=KMeansCluster(\tilde{D},k)$;
16. **for** 聚类结果 C 中所有的簇 c **do**
17. **for** c 中全部的实例 x **do**
18. $InsWeights_x=InsWeights_x+CalcWeights(c_j,LabelTotals,L)$;
19. **end for**
20. **end for**
21. **end for**
22. **for** $A+B$ 中所有实例 x **do**
23. $\tilde{y}=(InsWeights_x)$;
24. 将$(x,\tilde{y})$添加到 $\tilde{D}$;
25. **end for**
26. **return** $\tilde{D}$

算法 2　*CalcWeights*

输入:簇 c,标签分布向量 $\boldsymbol{v}$,标签集合 L。

输出:权重向量 $\boldsymbol{w}$。

1. $\boldsymbol{d}$ = 簇 c 中的标签分布
2. $u=1/|L|$;
3. $multiplier=\min(\log_{10}(\text{sizeof}(c)),2)$;
4. ***for*** $i=1$ to $|L|$ **do**
5. $w_i=multiplier\times\frac{d_i-u}{v_i}$;
6. **end for**
7. **return** $\boldsymbol{w}$

3　实验与结果分析

3.1　设置基准

依据数据质量和模型质量两个指标,将 MCNC 与 MV、PL、STC、CC、BMNC 进行比较。其中,MV 是指没有应用噪声纠正方法,仅使用多数投票算法来产生实例的集成标签。MV 的结果被作为基准与其余五种噪声纠正算法进行比较。数据质量的定义为:数据集中集成标签与真实标签相同的实例比例。模型质量定义为:在纠正后的数据集上训练目标分类器获得的分类精度。这里使用 C4.5 作为目标分类器。

本文在人群环境及其知识分析平台(CEKA)[28]上实现 MCNC 和 BMNC,使用 CEKA 平台现有的算法 MV、PL、STC 和 CC 的代码;使用怀卡托知识分析平台(WEKA)[29]的 C4.5(J48)代码。实验中的五种噪声纠正方法的设置如下:

(1) PL:C4.5 作为 PL 分类器。

(2) STC:用分类过滤器(CF)作为 STC 的过滤器,

纠正的噪声实例比例设置为 0.8，C4.5 作为 STC 分类器。

（3）CC：聚类次数 $a=10$，采用 k 均值聚类，k 值从 2 到实例数的一半不等。

（4）BMNC：CF 为过滤器，阈值 $\delta=0.95$，C4.5 为 BMNC 的分类器。

（5）MCNC：CF 为过滤器，阈值 $\delta=0.95$，聚类次数 $a=10$，采用 k 均值聚类为 CC 的聚类方法，k 值从 2 到实例数的一半不等。

另外，当 CF 作为 STC、BMNC 和 MCNC 的噪声过滤器时，需要设置一个参数 n（n 是对训练数据进行分区的数量）和用于过滤的分类器。在本文实验中，$n=10$，分类器同样是 C4.5。

3.2 模拟数据集和实验设置

在 22 个数据集上进行实验，表 1 展示了 22 个数据集的详细信息。为了模拟每个实例获得多标签集合的过程，隐藏了每个实例原本的真实标签，并使用 9 个模拟工人对每个实例进行标注。每个标注者的标注质量是 $p_j\in[0,1](j=1,2,\cdots,9)$，即对于每个工人来说，有 p_j 的概率给实例标注原本的真实标签，有 $1-p_j$ 的概率标注其他可能的标签。为了确定实验结果对于不同标注质量的稳定性，本文实验设置了两种不同的标注质量：

（1）在第一系列实验中，设置所有工人的标注质量为 0.6。即 $p_j=0.6(j=1,2,\cdots,9)$。

（2）在第二系列实验中，每个工人的标注质量均匀分布在[0.55,0.75]上，即 $p_j\in[0.55,0.75](j=1,2,\cdots,9)$。

表 1 数据集信息

数据集名称	分类属性数	数值属性数	实例数	类别数
balance-scale	0	4	625	3
blood	0	4	748	2
breast-w	9	0	683	2
credit-a	9	6	690	2
heart-c	13	7	303	5
heart-statlog	0	13	270	2
hepatitis	13	6	155	2
ionosphere	0	34	351	2
iris	0	4	150	3
lymph	15	3	148	4
seeds	0	7	210	3
segment-challenge	0	19	1 500	7
segment	0	19	2 310	7
sonar	0	60	208	2
wine	0	13	178	3
zoo	17	0	101	7

在每个实例获得 9 个工人标注的标签后，使用真值推理算法 MV 推理集成标签，然后应用五种噪声纠正算法识别并纠正集成标签中的噪声。在纠正后的数据集上计算数据质量，并在纠正后的数据集上训练目标分类器获得模型质量。所有实验结果都采用十折交叉验证得到，测试集不参与数据质量的计算。

3.3 实验结果

表 2 和表 3 给出了第一系列实验的结果，该实验中所有工人的标注质量都是相同的。表 2 展示了原始集成标签的标签质量和每个数据集分别应用五种噪声纠正算法后的标签质量。表 3 展示了应用不同纠正算法后的模型质量。

表 2 第一系列实验的标签质量结果（%）

数据集名称	MV	PL	STC	CC	BMNC	MCNC
balance-scale	78.42	**82.36**	77.46	81.03	76.35	81.39
blood	73.80	77.32	77.26	77.06	**79.55**	78.05
breast-w	71.89	93.83	86.94	95.09	94.66	**95.15**
credit-a	74.64	**86.57**	80.02	83.40	85.49	85.10
heart-c	75.91	82.40	75.91	82.69	80.53	**82.84**
hepatitis	74.84	80.15	78.71	84.37	81.43	**85.09**
ionosphere	75.78	81.10	78.06	88.10	88.19	**88.35**
iris	85.19	91.11	89.19	96.59	86.44	**97.63**
kr-vs-kp	72.97	95.16	87.25	82.36	**97.56**	86.19
labor	73.69	72.91	75.27	74.49	73.69	**77.02**
lymph	74.17	72.60	74.25	79.05	77.33	**81.38**
mushroom	72.85	98.51	91.64	94.77	**99.81**	95.62
seeds	89.42	91.85	91.59	94.23	85.40	**94.60**
segment-challenge	96.20	94.98	95.75	96.45	96.99	**98.53**
segment	96.20	95.68	95.50	96.89	97.62	**98.93**
sick	72.85	96.78	91.00	93.70	**97.99**	94.70
sonar	72.60	70.73	69.55	75.27	64.31	**76.44**
spambase	73.64	89.76	80.59	85.92	**90.62**	87.99
vote	72.18	93.10	85.64	89.71	**94.99**	91.52
vowel	97.24	72.83	83.85	96.33	88.68	**98.28**
wine	85.64	91.57	87.39	**96.51**	83.08	96.20
zoo	85.70	78.44	86.91	93.73	91.42	**95.49**

表 3　第一系列实验的模型质量结果(%)

数据集名称	MV	PL	STC	CC	BMNC	MCNC
balance-scale	75.38	77.61	74.55	77.12	71.04	**77.93**
blood	75.01	76.34	**77.15**	75.68	75.01	74.87
breast-w	90.93	93.11	92.38	92.82	90.93	**94.00**
credit-a	80.00	**84.93**	81.74	83.48	80.00	84.35
heart-c	70.67	74.86	70.67	**76.90**	70.67	76.58
hepatitis	67.21	75.00	78.08	**78.79**	67.21	78.75
ionosphere	79.20	78.90	80.63	82.30	79.20	**84.33**
iris	92.00	87.33	91.33	92.00	88.67	**92.67**
kr-vs-kp	93.62	94.77	**95.90**	89.02	93.62	94.65
labor	72.67	71.00	**74.00**	69.00	72.67	67.33
lymph	69.62	63.67	74.43	67.00	70.29	**75.05**
mushroom	99.04	98.52	99.08	99.79	99.04	**99.85**
seeds	88.57	89.05	90.95	90.48	**92.86**	90.48
segment-challenge	94.47	93.13	94.07	94.20	94.40	**94.87**
segment	96.23	94.07	94.68	94.98	96.32	**96.84**
sick	96.92	**97.06**	96.90	95.57	96.92	96.61
sonar	54.38	63.40	59.64	65.83	54.38	**67.33**
spambase	88.09	**89.20**	86.76	84.92	88.09	86.63
vote	94.03	93.34	**94.26**	89.68	94.03	92.67
vowel	78.48	59.39	71.62	78.48	**80.20**	78.69
wine	84.90	86.57	85.42	**92.78**	85.46	89.97
zoo	91.18	76.27	91.18	**95.00**	91.18	94.00

从表 2 中可以看出,所有的噪声纠正方法都可以在大部分数据集上提升标签质量。MCNC 在 13 个数据集上的效果最好,即在 13 个数据集上提升标签质量最多(例如:iris、labor 和 segment 等)。其次是 BMNC,在 6 个数据集上提升标签质量最多。后面分别是 PL 和 CC,分别在 2 个和 1 个数据集上取得最好的效果。而 STC 没有在任何一个数据集上取得最好的效果。从表 3 可以看出,在一些情况下,提升标签质量可以提高目标分类器的性能。MCNC 在 9 个数据集上性能最优,其次是 STC 和 CC 的 4 个,PL 在 3 个数据集表现最好,BMNC 只有 2 个。

表 4 和表 5 给出了第二系列实验的结果,该实验中所有工人的标注质量均匀分布在[0.55,0.75]之间。表 4 展示了原始集成标签的标签质量和每个数据集分别应用五种噪声纠正算法后的标签质量。表 5 展示了应用不同纠正算法后的模型质量。

表 4　第二系列实验的数据质量结果(%)

数据集名称	MV	PL	STC	CC	BMNC	MCNC
balance-scale	86.36	84.71	86.33	**86.54**	80.92	86.49
blood	81.55	76.49	80.26	80.39	80.05	**82.80**
breast-w	82.72	**97.27**	91.54	96.47	96.14	96.76
credit-a	83.19	86.70	84.65	85.73	88.23	**88.49**
heart-c	82.84	83.75	82.84	84.86	83.21	**85.85**
hepatitis	84.51	86.73	84.73	88.17	83.87	**89.46**
ionosphere	82.34	86.64	82.59	89.33	89.65	**90.79**
iris	93.78	94.07	94.37	96.74	92.44	**96.96**
kr-vs-kp	82.38	95.40	93.53	88.33	**98.75**	93.20
labor	84.21	69.00	78.59	86.16	79.14	**88.11**
lymph	80.03	79.88	79.96	82.96	76.43	**84.16**
mushroom	83.09	98.52	95.76	98.81	**100.0**	99.03
seeds	92.33	90.69	92.12	94.07	87.78	**94.44**
segment-challenge	97.61	94.58	96.41	96.50	97.00	**98.70**
segment	98.22	95.58	96.94	96.83	97.77	**98.99**
sick	83.88	**98.22**	94.47	95.80	98.20	97.45
sonar	82.69	75.70	77.14	82.10	75.00	**84.67**
spambase	83.13	**93.09**	86.34	89.82	92.82	92.58
vote	83.68	95.22	90.93	91.72	**95.45**	94.05
vowel	98.14	73.79	85.19	96.04	87.99	**98.51**
wine	91.01	92.51	92.07	96.76	86.52	**96.94**
zoo	90.32	77.13	89.99	**93.84**	88.01	93.73

表 5　第二系列实验的模型质量结果(%)

数据集名称	MV	PL	STC	CC	BMNC	MCNC
balance-scale	77.89	78.05	79.18	**79.50**	78.53	78.54
blood	76.74	76.47	76.88	77.41	76.74	**78.08**
breast-w	92.53	**95.76**	93.12	95.02	92.53	94.87
credit-a	86.09	85.36	84.93	85.22	86.09	**86.96**
heart-c	70.00	75.56	70.00	78.53	70.00	**78.84**
hepatitis	82.67	**85.25**	82.13	84.00	82.67	82.75
ionosphere	80.31	84.90	84.03	**86.29**	80.31	84.33
iris	92.67	**94.67**	92.67	**94.67**	92.67	94.00
kr-vs-kp	97.93	95.21	**98.09**	94.02	97.93	97.90
labor	77.00	70.67	70.67	**78.33**	77.00	74.33
lymph	72.38	71.05	71.05	71.05	69.62	**74.43**
mushroom	99.93	98.52	99.98	99.98	99.93	**100.0**
seeds	90.00	88.57	89.52	88.57	**90.95**	89.05
segment-challenge	94.60	92.60	94.13	93.67	**95.27**	94.67

续表 5

数据集名称	MV	PL	STC	CC	BMNC	MCNC
segment	96.75	94.24	95.41	95.50	**96.93**	96.75
sick	97.83	97.85	**98.09**	95.94	97.83	97.51
sonar	61.50	63.48	59.57	66.33	61.50	**70.29**
spambase	89.59	**91.42**	89.29	86.98	89.59	89.46
vote	94.72	**95.41**	94.73	92.19	94.72	94.95
vowel	80.20	60.51	69.60	76.77	**81.41**	76.67
wine	88.20	88.20	89.93	90.42	89.35	**90.46**
zoo	89.09	74.18	**90.09**	88.00	87.09	88.09

从表 4 可以看出，MCNC 的性能依然是最优的，在 14 个数据集上取得最好效果，BMNC 和 PL 在 3 个数据集上性能最优，CC 只有 2 个，STC 依旧没有突出性能的数据集。表 5 显示，MCNC 在 7 个数据集上性能最优，PL 有 5 个，CC 和 BMNC 都有 4 个，STC 仅有 3 个。

结合两个系列的实验结果，在大部分数据集上，改善众包数据的标签质量，可以在一定程度上提升相应的目标模型质量。但不同位置的实例对于模型建立的贡献度是不同的，直观而言，分类决策面的边界数据点的贡献度要高于类内部的数据点，所以数据标签质量的提升并不必然导致目标模型质量的提升。

通过上面两个系列的实验，相比较 PL、STC、CC 和 BMNC，本文方法 MCNC 更能有效地提升众包数据的标签质量和目标模型质量。

4 结 语

本文提出一种新的针对众包学习的标签噪声纠正方法 MCNC。本文方法使用了真值推理阶段的信息来监督噪声的识别，使用了无视原本噪声等级的基于聚类的算法进行噪声纠正。相对于被比较的各种方法，MCNC 能够有效地提升标签质量和模型质量。

后续工作将针对提升数据标签质量并非一定提升目标模型质量的现象，研究分类决策面的边界数据点的标签噪声纠正。期望可以通过提升分类决策面的边界数据点标签质量，使得目标模型质量获得较大提升。然而哪些实例更有可能是分类决策面的边界点数据，对模型建立有更高的贡献度，是需要仔细考虑的问题。同时，基于聚类的集成方法对计算资源消耗较大，运行时间较长，后续会对该方法进一步优化，以减少运行时间。

参 考 文 献

[1] Sáez J A, Galar M, Luengo J, et al. Analyzing the presence of noise in multi-class problems: Alleviating its influence with the One-vs-One decomposition[J]. Knowledge and Information Systems, 2014, 38: 179 – 206.

[2] Ipeirotis P G, Provost F, Sheng V, et al. Repeated labeling using multiple noisy labelers[J]. Social Science Electronic Publishing, 2014, 28(2): 402 – 441.

[3] Raykar V C, Yu S, Zhao L H, et al. Learning from crowds [J]. Journal of Machine Learning Research, 2010, 11(2): 1297 – 1322.

[4] Demartini G, Difallah D E, Cudré-Mauroux P. ZenCrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking[C]//21st International Conference on World Wide Web, 2012: 469 – 478.

[5] Sheng V S. Simple multiple noisy label utilization strategies [C]//2011 11th IEEE International Conference on Data Mining, 2011: 635 – 644.

[6] Tian T, Jun Z. Max-Margin majority voting for learning from crowds[C]//28th International Conference on Neural Information Processing Systems, 2015.

[7] Zhang H, Jiang L, Xu W. Differential evolution-based weighted majority voting for crowdsourcing[C]//15th Pacific Rim International Conference on Artificial Intelligence, 2018.

[8] Zhao L, Sukthankar G, Sukthankar R. Incremental relabeling for active learning with noisy crowdsourced annotations [C]//2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing, 2011: 728 – 733.

[9] Fang M, Zhu X, Li B, et al. Self-Taught active learning from crowds [C]//IEEE International Conference on Data Mining, 2013: 858 – 863.

[10] Kajino H, Tsuboi Y, Sato I, et al. Learning from crowds and experts[J]. Transactions of the Japanese Society for Artificial Intelligence, 2013, 28(3): 243 – 248.

[11] Hu Q, He Q, Huang H, et al. Learning from crowds under experts' supervision [J]. Lecture Notes in Computer Science, 2014, 8443: 200 – 211.

[12] Shu Z, Sheng V S, Zhang Y, et al. Integrating active learning with supervision for crowdsourcing generalization[C]// 2015 IEEE 14th International Conference on Machine Learning and Applications ICMLA), 2015: 232 – 237.

[13] Muhammadi J, Rabiee H R, Hosseini A. A unified statistical framework for crowd labeling[J]. Knowledge and Information Systems, 2015, 45(2): 271 – 294.

[14] Frenay B, Verleysen M. Classification in the presence of label noise: A survey[J]. IEEE Transactions on Neural Networks and Learning Systems, 2014, 25(5): 845 – 869.

(下转第 37 页)

Application to ball bearing fault detection[J]. International Journal of Control Automation & Systems, 2017, 15(2): 1 – 12.

[5] Jaffel I, Taouali O, Harkat M F, et al. Kernel principal component analysis with reduced complexity for nonlinear dynamic process monitoring[J]. International Journal of Advanced Manufacturing Technology, 2016, 88(9 – 12): 1 – 15.

[6] Adedigba S A, Khan F, Yang M. Dynamic failure analysis of process systems using principal component analysis and bayesian network[J]. Industrial & Engineering Chemistry Research, 2017, 56(8): 2094 – 2106.

[7] Ge Z Q, Yang C J, Song Z H. Improved kernel PCA-based monitoring approach for nonlinear processes[J]. Chemical Engineering Science, 2009, 64(9): 2245 – 2255.

[8] Schölkopf B, Smola A, Müller K. Nonlinear component analysis as a kernel eigenvalue problem[J]. Neural Computation, 1998, 10(5): 1299 – 1319.

[9] Wu F, Yin S, Karimi H R. Fault detection and diagnosis in process data using support vector machines[J]. Journal of Applied Mathematics, 2014(8): 1 – 9.

[10] Shen L, Wang H, Xu L D, et al. Identity management based on PCA and SVM[J]. Information Systems Frontiers, 2016, 18(4): 711 – 716.

[11] Vapnik V N. The nature of statistical learning theory[M]. Springer, 1999.

[12] 郭金玉,刘玉超,李元. 一种基于改进局部熵 PCA 的工业过程故障检测方法[J]. 高校化学工程学报, 2019, 33(4): 922 – 932.

[13] Guo J, Wang X, Li Y. Fault detection based on improved local entropy locality preserving projections in multimodal processes[J]. Journal of Chemometrics, 2019, 33(3): 3116.

[14] Downs J J, Vogel E F. A plant-wide industrial process control problem[J]. Computers and Chemical Engineering, 1993, 17(3): 245 – 255.

[15] Mcavoy T J, Ye N. Base control for the Tennessee Eastman problem[J]. Computers & Chemical Engineering, 1994, 18(5): 383 – 413.

[16] Lee G, Han C, Yoon E S. Multiple-fault diagnosis of the Tennessee Eastman process based on system decomposition and dynamic PLS[J]. Industrial & Engineering Chemistry Research, 2004, 43(25): 8037 – 8048.

[17] Yin S, Ding S X, Haghani A, et al. A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process[J]. Journal of Process Control, 2012, 22(9): 1567 – 1581.

[18] Ma H, Hu Y, Shi H. Fault detection and identification based on the neighborhood standardized local outlier factor method[J]. Industrial & Engineering Chemistry Research, 2013, 52(6): 2389 – 2402.

(上接第 12 页)

[15] Duan Y, Wu O. Learning with auxiliary less-noisy labels[J]. IEEE Transactions on Neural Network and Learning Systems, 2017, 28(7): 1716 – 1721.

[16] Miao Q, Cao Y, Xia G, et al. RBoost: Label noise-robust boosting algorithm based on a nonconvex loss function and the numerically stable base learners[J]. IEEE Transactions on Neural Networks and Learning Systems, 2016, 27(11): 2216 – 2228.

[17] Varon C, Alzate C, Suykens J A K. Noise level estimation for model selection in Kernel PCA denoising[J]. IEEE Transactions on Neural Networks and Learning Systems, 2015, 26(11): 2650 – 2663.

[18] Sun J W, Zhao F Y, Wang C J, et al. Identifying and correcting mislabeled training instances[C]//Future Generation Communication and Networking, 2008: 244 – 250.

[19] Ekambaram R, Fefilatyev S, Shreve M, et al. Active cleaning of label noise[J]. Pattern Recognition, 2015, 51: 463 – 480.

[20] Malossini A, Blanzieri E, Ng R T. Detecting potential labeling errors in microarrays by data perturbation[J]. Bioinformatics, 2006, 22(17): 2114 – 2121.

[21] 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, 2016, 28(4): 1080 – 1085.

[22] Li C, Sheng V S, Jiang L, et al. Noise filtering to improve data and model quality for crowdsourcing[J]. Knowledge-Based Systems, 2016, 107: 96 – 103.

[23] Nicholson B, Zhang J, Sheng V S, et al. Label noise correction methods[C]//2015 IEEE International Conference on Data Science and Advanced Analytics(DSAA), 2015: 1 – 9.

[24] Teng C M. Correcting noisy data[C]//16th International Conference on Machine Learning(ICML 1999), 1999: 239 – 241.

[25] Triguero I, Sáez J A, Luengo J, et al. On the characterization of noise filters for self-training semi-supervised in nearest neighbor classification[J]. Neurocomputing, 2014, 132: 30 – 41.

[26] Zhang J, Sheng V S, Wu J, et al. Improving label quality in crowdsourcing using noise correction[C]//24th ACM International on Conference on Information and Knowledge Management, 2015.

[27] 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.

[28] Zhang J, Sheng V S, Nicholson B A, et al. CEKA: A tool for mining the wisdom of crowds[J]. Journal of Machine Learning Research, 2015, 16(1): 2853 – 2858.

[29] Witten I H, Frank E. 数据挖掘:实用机器学习工具与技术[M]. 北京:机械工业出版社, 2005.