一种基于自训练的众包标记噪声纠正算法

杨 艺 [1] 蒋良孝 [1, 2] 李超群 [3]

摘 要 针对众包标记经过标记集成后仍然存在噪声的问题, 提出了一种基于自训练的众包标记噪声纠正算法 (Self-training-based label noise correction, STLNC). STLNC 整体分为 3 个阶段: 第 1 阶段利用过滤器将带集成标记的众包数据集分为噪声集和干净集. 第 2 阶段利用加权密度峰值聚类算法构建数据集中低密度实例指向高密度实例的空间结构关系. 第 3 阶段首先根据发现的空间结构关系设计噪声实例选择策略; 然后利用在干净集上训练的集成分类器对选择的噪声实例按照设计的实例纠正策略进行纠正, 并将纠正后的实例加入到干净集, 再重新训练集成分类器; 重复实例选择与纠正过程直到噪声集中所有的实例被纠正; 最后用最后一轮训练得到的集成分类器对所有实例进行纠正. 在仿真标准数据集和真实众包数据集上的实验结果表明 STLNC 比其他 5 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, 自训练, 集成标记, 标记噪声, 噪声纠正

引用格式 杨艺, 蒋良孝, 李超群. 一种基于自训练的众包标记噪声纠正算法. 自动化学报, 2023, **49**(4): 830−844

DOI 10.16383/j.aas.c210051

A Self-training-based Label Noise Correction Algorithm for Crowdsourcing

YANG Yi[1] JIANG Liang-Xiao[1, 2] LI Chao-Qun[3]

Abstract In order to solve the problem that a certain level of label noise exists in integrated labels obtained by label integration algorithms, this paper proposes a self-training-based label noise correction (STLNC) algorithm for crowdsourcing. There are three stages in STLNC. At the first stage, STLNC employs a filter to get a clean set and a noisy set. At the second stage, the weighted density peak clustering algorithm is used to construct the spatial structure relationship between low-density instances and high-density instances in the dataset. At the third stage, a noise instance selection strategy is at first designed according to the found spatial structure relationship. Then, these selected noise instances are corrected by the ensemble classifier trained on the clean set according to the designed instance correction strategy, and the corrected instances are added into the clean set and the ensemble classifier is retrained. The process of instance selection and correction is repeated until all noise instances are corrected. Finally, the ensemble classifier trained from the last round is used to correct all the instances. Experimental results on both simulated benchmark datasets and real-world crowdsourced datasets show that STLNC significantly outperforms other five state-of-the-art noise correction algorithms in team of the noise ratio and the model quality.

Key words Crowdsourcing learning, self-training, integrated labels, label noise, noise correction

Citation Yang Yi, Jiang Liang-Xiao, Li Chao-Qun. A self-training-based label noise correction algorithm for crowdsourcing. *Acta Automatica Sinica*, 2023, **49**(4): 830−844

随着计算机技术和互联网技术的飞速发展, 当今社会进入了大数据时代, 数据的重要性也变得越来越高. 而与此相关的人工智能领域, 例如目标检测[1]、图像识别[2] 和语音识别[3] 等, 对数据的需求也在不断提高. 但是, 这些图像和语音类数据需要标注的数据量巨大, 采用传统的专家标注方法已经不能满足需求.

近年来, 随着 AMT (Amazon mechanical turk)[1]、CrowdFlower[2] 和 Clickworker[3] 等众包平台的出现, 众包技术为获取大量数据标记提供了一种经济、高效的方式. 众包学习也因此成为了一个新

收稿日期 2021-01-18 录用日期 2021-05-12
Manuscript received January 18, 2021; accepted May 12, 2021
国家自然科学基金联合基金 (U1711267), 中央高校基本科研业务费专项资金 (CUGGC03) 资助
Supported by National Natural Science Foundation of China (U1711267) and Fundamental Research Funds for the Central Universities (CUGGC03)
本文责任编委 胡清华
Recommended by Associate Editor HU Qing-Hua
1. 中国地质大学 (武汉) 计算机学院 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 (武汉)) 武汉 430074 3. 中国地质大学 (武汉) 数学与物理学院 武汉 430074
1. School of Computer Science, China University of Geosciences (Wuhan), Wuhan 430074 2. Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences (Wuhan), Wuhan 430074 3. School of Mathematics and Physics, China University of Geosciences (Wuhan), Wuhan 430074

[1] http://www.mturk.com
[2] http://www.crowdflower.com
[3] http://www.clickworker.com

兴的研究领域. 在众包学习中, 首先通过在线的平台雇佣多个众包工人, 对每个实例进行标注, 获得该实例的多噪声标记集. 然后通过标记集成算法, 从每个实例的多噪声标记集中推理出一个合理的集成标记. 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: MV (Majority voting) 算法[4]、ZC (ZenCrowd) 算法[5]、MNLDP (Multiple noisy label distribution propagation) 算法[6]、M3V (Max-margin majority voting) 算法[7]、QS-LFC (Quality-sensitive learning from crowds) 算法[8].

然而, 众包工人在专业知识水平、打标积极性和对评价指标的理解等方面的差异性, 导致数据标注的质量普遍偏低. 因此标记集成算法得到的集成标记中仍然存在一定比例的噪声, 即实例的集成标记与真实标记不一致. 标记噪声的存在会损害数据集的标记质量, 而具有高标记质量的数据集在相关研究和应用技术中又是至关重要的. 所以, 对标记集成后的数据集通过众包标记噪声纠正算法进行纠正从而提高数据集的标记质量具有重要的研究意义.

为了降低标记噪声的影响, 提高标记集成后的数据集的标记质量, 目前已经有研究者在众包标记噪声纠正方向做了一些工作, 主要可分为 3 类:

1) 基于监督学习的众包标记噪声纠正算法. Nicholson 等[9] 提出了适用于众包领域的标记噪声纠正算法 PL (Polishing labels). PL 算法通过将数据集分成 10 个子集, 并在每个子集上训练一个分类器, 由生成的 10 个分类器对每个实例进行分类投票, 最终将得票最高的标记赋予实例. 而 Xu 等[10] 结合重抽样的思想, 提出了一种基于重抽样的众包标记噪声纠正算法 (Resampling-based noise correction, RNC). RNC 通过在干净集和噪声集上按照比例多次重抽样训练多个分类器, 再用得到的多个分类器对整个数据集进行纠正, 从而提高数据集的标记质量. 除了以上两种众包标记纠正算法, 部分研究者还将研究的重点聚焦于监督学习领域知识和众包数据集信息的不确定性方法, 其中的代表性算法有 AVNC (Adaptive voting noise correction) 算法[11]、BMNC (Between-class margin-based noise correction) 算法[12] 以及 CENC (Cross-entropy-based noise correction) 算法[13]. AVNC 算法提出了众包领域的噪声纠正框架, 利用众包服务提供的标记信息, 估计了众包工人的质量, 并进一步评估了数据集中含有噪声的比例. 然后对实例的噪声等级进行了排序, 最终通过集成模型对被认定为噪声的实例进行纠正. BMNC 算法则是利用了众包数据集中实例的多噪声标记集信息, 评估每个实例的集成标记的置信程度, 从而更加精准地过滤出噪声实例, 并通过最终得到的干净集训练分类器对噪声实例进行纠正. 而 CENC 算法同样利用了众包数据集中实例的多噪声标记集, 计算多噪声标记集的信息熵去评估集成标记的置信程度, 然后在得到的干净集上训练多个分类器对噪声集实例进行预测, 进一步利用交叉熵的思想去衡量噪声集中实例标记的真实分布和预测分布的相似度, 从而实现对噪声实例的纠正, 提高数据集的标记质量.

2) 基于无监督学习的众包标记噪声纠正算法. Nicholson 等[9] 提出了一种基于聚类的众包标记噪声纠正算法 CC (Cluster-based correction). 聚类是无监督学习技术, 在噪声处理领域的优势是不依赖实例的类标记. 这种优势的存在使得该方法在性能上往往优于基于监督学习的算法, 但是 CC 算法在时间效率方面则较差. CC 的核心思想是进行多次聚类算法, 在每次聚类执行时, 对每个簇中的实例计算并赋予相同的权重, 权重反映了实例属于不同类别的可能性. 最终根据每个实例的最大权值对实例进行重新标注, 从而提高数据集的标记质量.

3) 基于半监督学习的众包标记噪声纠正算法. Nicholson 等[9] 提出的 STC (Self-training correction) 算法则是受自训练过程的启发, 首先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分类器用于对噪声集中的实例进行预测打标, 再选取每类中打标置信度最高的噪声实例, 将预测标记赋予实例并加入干净集中, 重复上述步骤直到纠正的噪声实例达到设定的阈值. 通过以上步骤, 该算法提高了数据集的标记质量和模型精度.

在上文提到的 3 类众包标记噪声纠正算法中, 基于监督学习的方法训练分类器估计实例的集成标记质量, 并对噪声实例进行纠正; 基于无监督学习的方法不依赖数据集的集成标记, 通过数据的特征估计实例所属某类的权重; 基于半监督学习的方法是将干净集实例作为已标记实例来训练分类器, 对噪声集实例进行重新标注的方法. 在基于半监督学习的方法中, STC 算法因为简单有效, 不需要特定的假设条件而被广泛应用. 但通过对 STC 算法的分析, 本文发现该算法仍然存在 3 个方面的不足: 1) 当用于训练初始分类器的干净集不能很好表示整个数据空间, 难以反映数据的分布时, STC 算法得到分类器的效果较差. 从而较多错误纠正的实例加入干净集, 误差的影响在循环中不断扩大, 导致 STC 算法效果受损; 2) 经过过滤得到的干净集仍然存在噪声实例, 而 STC 算法是在干净集上训练单个分类器用于纠正噪声集, 因此训练的分类器纠正

效果不佳; 3) STC 算法是基于半监督学习领域的思想, 未充分利用众包数据集所含的标记信息, 难以取得更好的众包标记噪声纠正效果.

针对 STC 算法存在的不足, 本文提出了一种基于自训练的众包标记噪声纠正算法 (Self-training-based label noise correction, STLNC). 本文同时使用仿真标准数据集和真实众包数据集进行了实证研究, 结果表明在噪声比和模型质量两个度量指标上, STLNC 算法比其他五种最先进的标记噪声纠正算法表现更好.

1 一种基于自训练的众包标记噪声纠正算法

1.1 背景知识

在一个众包系统中, N 个实例由 J 个工人打标后, 获得了众包数据集 $D=\{(\boldsymbol{x}_i, L_i)\}_{i=1}^{N}$, 其中 $L_i=\{l_{ij}\}_{j=1}^{J}$ 为每个实例的多噪声标记集, l_{ij} 表示实例 $\boldsymbol{x}_i$ 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$\{+, 0, -\}$, 其中的值分别表示众包工人对实例打正标, 未给出标记和打负标. 众包数据集 D 中的每个实例 $\boldsymbol{x}_i$ 在经过标记集成算法推理得到集成标记 $\hat{y}_i$ 后, 获得了带有集成标记的众包数据集 $D'=\{(\boldsymbol{x}_i, \hat{y}_i)\}_{i=1}^{N}$. 而数据集 D' 正是众包标记噪声纠正工作的起点.

1.2 STLNC 算法

上文分析了 STC 算法的不足, 为了提高 STC 算法的性能, 本文提出的 STLNC 算法利用加权密度峰值聚类算法构建数据集中低密度实例指向高密度实例的空间结构关系, 以便更好地指导噪声集实例的选择. 同时, 将 STC 中的单个分类器替换为集成分类器, 在纠正阶段利用设计的纠正策略对选择的噪声实例进行纠正, 降低了纠正的误差, 提高了数据集的标记质量和模型精度, 整个算法的框架如图 1 所示.

从图 1 可以看出, STLNC 算法工作的起点是带有集成标记的数据集 D', 该数据集的集成标记是通过标记集成算法推理获得的. 在获得数据集 D' 后, STLNC 算法可分为 3 个阶段: 1) 噪声过滤; 2) 构建空间结构关系; 3) 实例选择与纠正. 在噪声过滤阶段, 采用过滤器对 D' 进行过滤, 得到干净集 D'_c 和噪声集 D'_n; 在构建空间结构关系阶段, 利用加权密度峰值聚类构建数据集中低密度实例指向高密度实例的空间结构关系; 在实例选择与纠正阶段, 根据获得的干净集和噪声集以及空间结构关系设计实例选择策略, 在每次循环中选择最适合纠正的噪声实例. 再利用干净集 D'_c 训练集成分类器 M, 对选择的噪声实例通过设计的实例纠正策略进行纠正, 然后加入到 D'_c 中. 循环执行实例选择和纠正步骤, 直到所有噪声实例被选择并纠正, 获得最后一轮训练的集成分类器 M'. 最后采用集成分类器 M' 通过多数投票 (Majority voting) 的方式对 D' 所有实例进行纠正, 得到最终纠正后的数据集 D'_{correct}.

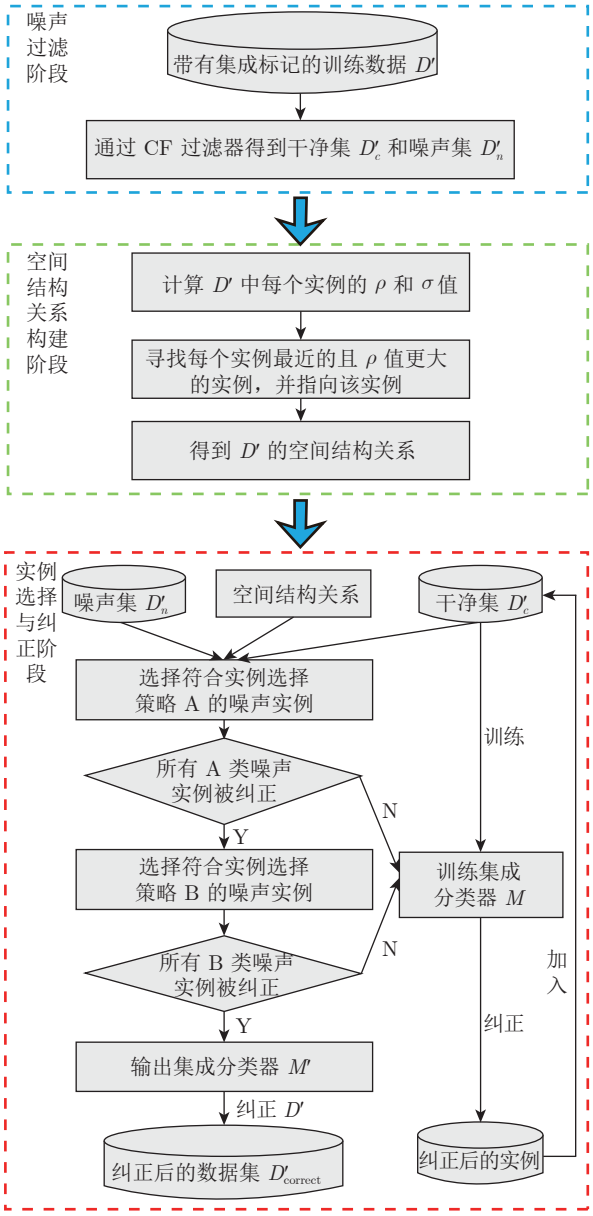


图 1 STLNC 算法的框架

Fig. 1 Framework of STLNC

在整个 STLNC 算法流程中, 空间结构关系的构建以及实例选择和纠正策略的设计是本算法的关键, 本文将在后续内容中详细介绍.

1.2.1　构建空间结构关系

为了解决精确选择噪声实例的问题, 受到 Wu 等[14] 的启发, STLNC 算法利用了加权密度峰值聚类算法构建数据集中低密度实例指向高密度实例的空间结构关系, 从而指导噪声实例选择过程.

密度峰值聚类的主要思想基于以下两点: 1) 聚类中心的密度大于其邻居的密度, 也就是被低密度实例围绕; 2) 聚类中心与更高密度的实例之间的距离较远, 即簇间距离较大. 对于众包数据集 $D^{'}$ 中每一个实例 $\boldsymbol{x}_i$, 它的局部密度 ρ_i 定义为

$$\rho_i = \sum_{j,\, j\neq i} I\left(d_{ij} < d_c\right) \tag{1}$$

其中, $I(\cdot)$ 为指示函数, $d_{ij} < d_c$ 值取 1, 否则为 0. d_{ij} 为实例 $\boldsymbol{x}_i$ 与 $\boldsymbol{x}_j$ 之间的距离, 一般为欧氏距离. d_c 为截断距离, 具体定义为

$$d_c = d_{C_N^2 \times s} \tag{2}$$

其中, $d_{C_N^2 \times s} \in [d_1, d_2, \cdots, d_{C_N^2}]$, 该集合包含每对实例之间的距离, 并按照升序排序. $d_{C_N^2 \times s}$ 代表该集合中第 $C_N^2 \times s$ 个距离, s 为原始密度峰值聚类算法给出的经验参数, 本文取值为 0.2.

在获取到局部密度 ρ_i 之后, 密度峰值聚类算法需要进一步得到实例 $\boldsymbol{x}_i$ 的 δ_i 值, δ_i 定义为: 1) 对于非最高密度实例 $\boldsymbol{x}_i$, 计算实例 $\boldsymbol{x}_i$ 与较高密度最近邻的距离; 2) 对于最高密度实例 $\boldsymbol{x}_i$, 计算实例 $\boldsymbol{x}_i$ 和最远实例之间的距离. 具体为

$$\delta_i = \begin{cases} \max_j\left(d_{ij}\right), & \forall j,\ \rho_i \geq \rho_j \\ \min_{j:\rho_i<\rho_j}\left(d_{ij}\right), & \text{其他} \end{cases} \tag{3}$$

在计算出所有实例的局部密度 ρ_i 和距离 δ_i 之后, 局部密度 ρ_i 和距离 δ_i 均相对较大的实例将作为簇的中心, 剩下的实例被归于密度较高的最近邻所属的簇, 从而得到最终的聚类结果.

在本文中, 为了精确构建众包数据集实例的空间结构关系, 我们将众包数据集的信息引入密度峰值聚类算法的距离计算中. 对于每个众包数据集中的实例, 实例所带有的多噪声标记集以及众包数据集的工人质量是能够反映实例之间相关关系的重要属性. 因此本文在计算实例 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 之间的距离 $d_{ij} = dist\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)$ 时, STLNC 算法通过使用实例的多噪声标记集衡量实例之间的相关关系, 具体定义如下:

$$d_{ij} = dist\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = w_{ij}\sqrt{\sum_{k=1}^{K}\left(x_{ik} - x_{jk}\right)^2} \tag{4}$$

其中, K 是特征维度, w_{ij} 表示工人对实例 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 的打标相似度. 为了利用实例的多噪声标记集计算实例 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 的打标相似度, 首先需要根据实例的噪声标记集 $L_i = \{l_{ij}\}_{j=1}^{J}$, 构建每个实例的标记分布, 对二类数据集, 实例具体的标记分布为 $\boldsymbol{P}_i = \left\{p_i^+, p_i^-\right\}$, 其中 p_i^+ 代表的是所有工人对实例 $\boldsymbol{x}_i$ 打标为正类的概率, p_i^- 代表的是所有工人对实例 $\boldsymbol{x}_i$ 打标为负类的概率. 具体为

$$\begin{cases} p_i^+ = \dfrac{Num_i(+) + 1}{Num_i(+) + Num_i(-) + 2} \\ p_i^- = \dfrac{Num_i(-) + 1}{Num_i(+) + Num_i(-) + 2} \end{cases} \tag{5}$$

其中, $Num_i(+)$ 为实例 $\boldsymbol{x}_i$ 多噪声标记集中正类标记的数目, $Num_i(-)$ 为实例 $\boldsymbol{x}_i$ 多噪声标记集中负类标记的数目. 为了满足后续标记分布之间关系计算的需要, 式 (5) 使用了拉普拉斯纠正[15]. 在本算法中, 使用标记分布的好处是能够使用数值显式地表示工人打标的分布, 以便精确估计两个实例之间打标的相似程度. 因此在获得每个实例的标记分布之后, 我们采用 KL (Kullback-Leibler) 散度[16] 来计算两个实例标记分布的近似程度, KL 散度越小则说明工人对两个实例打标的情况越接近, 因此 w_{ij} 也越小, 具体为

$$\begin{aligned} w_{ij} = \frac{1}{2}\left(KL\left(\boldsymbol{P}_i||\boldsymbol{P}_j\right) + KL\left(\boldsymbol{P}_j||\boldsymbol{P}_i\right)\right) = \\ \frac{1}{2}\left(\sum_{m\in\{+,-\}} p_i(m)\cdot\log\frac{p_i(m)}{p_j(m)} + \right. \\ \left. \sum_{m\in\{+,-\}} p_j(m)\cdot\log\frac{p_j(m)}{p_i(m)}\right) \end{aligned} \tag{6}$$

其中, $p_i(m)$ 表示实例 $\boldsymbol{x}_i$ 标记属于 m 类的概率.

本文在根据实例之间的打标相似度 w_{ij} 得到 d_{ij} 后, 继续进行密度峰值聚类过程. 直到计算实例的 δ_i 值时, 会为每个实例 $\boldsymbol{x}_i$ 找到具有更高密度的最近邻实例 $\boldsymbol{x}_j$. 根据 Wu 等[14] 的证明, 通过实例 $\boldsymbol{x}_i$ 指向实例 $\boldsymbol{x}_j$ 可以发现整个数据空间的真实结构. 这种由低密度实例 $\boldsymbol{x}_i$ 指向高密度实例 $\boldsymbol{x}_j$ 的空间结构关系不会因为实例的标记改变而发生变化, 是一种稳定的结构关系, 能够指导我们发现密度更高更具有代表性的实例. 因此, 本文执行加权密度峰值聚类步骤计算所有实例的 δ_i, 从而也获得了整个众包数据集的空间结构关系.

1.2.2　实例选择与纠正策略

在通过加权密度峰值聚类算法构建数据集中低

密度实例指向高密度实例的空间结构关系之后, 本文结合噪声过滤阶段得到的干净集和噪声集, 设计了两种噪声实例选择策略. 首先, 本文定义了两类实例点, 当实例 $\boldsymbol{x}_i$ 指向实例 $\boldsymbol{x}_j$, 那么定义 $\boldsymbol{x}_i$ 为 $\boldsymbol{x}_j$ 的"previous"点, $\boldsymbol{x}_j$ 为 $\boldsymbol{x}_i$ 的"next"点.

1) 实例选择策略 A: 在整个数据空间中, 搜寻所有"previous"点为干净集实例, "next"点为噪声集实例的实例对, 将每个实例对中的噪声集实例作为纠正的对象;

2) 实例选择策略 B: 在整个数据空间中, 搜寻所有"previous"点为噪声集实例, "next"点为干净集实例的实例对, 将每个实例对中的噪声集实例作为纠正的对象.

在实例选择与纠正阶段, 首先根据实例选择策略 A 选择噪声实例, 当没有满足条件的噪声实例时, 再使用实例选择策略 B 选择实例, 直到噪声集实例全部纠正并加入干净集. 根据第 1.2.1 节的说明, 本文构建空间结构关系目的是为了发现实例之间的真实结构, 指导算法寻找更高密度更具代表性的实例, 从而训练出更加可靠的分类器. 因此本文先执行了实例选择策略 A 来寻找具有更高密度的噪声实例, 这些高密度噪声实例更具代表性. 将这些高密度噪声实例先进行选择并纠正能够使扩充后的干净集更好的代表整个数据空间, 使训练得到的分类器更加可靠, 提高标记纠正的效果. 在执行完实例选择策略 A 直到没有满足条件的噪声实例后, 算法将训练得到一个可靠的分类器. 为了进一步扩充干净集, 本文随后执行了实例选择策略 B, 对剩余的低密度噪声实例进行选择, 从而完成了整个实例选择的过程.

解决噪声实例选择的问题后, 剩下的问题就是如何设计实例纠正策略, 提高纠正的准确性. 在 STLNC 算法中, 为了提高纠正的准确性, 本文将用于标记纠正的单个分类器替换为由 3 个异质的分类器构成的集成分类器. 同时根据第 1.2.1 节得到的每个实例的标记分布 $\boldsymbol{P}_j$, 计算每个实例多噪声标记集的信息熵值 $Entropy\,(\boldsymbol{P}_i)$, 从而估计该实例集成标记的置信度 e_i. 当 e_i 越小时, 说明工人打标的一致性越高, 则集成标记的置信度越高; 反之则集成标记的置信度越低. e_i 的具体计算为

$$e_i = Entropy\,(\boldsymbol{P}_i) = -\sum_{m\in\{+,-\}} p_i(m)\log p_i(m) \quad (7)$$

其中, $p_i(m)$ 表示实例 $\boldsymbol{x}_i$ 标记属于 m 类的概率.

本文结合上述集成分类器和众包数据集标记信息, 设计了 STLNC 的实例纠正策略:

1) 当实例 $\boldsymbol{x}_i$ 的集成标记置信度 e_i 大于设定的阈值 T, 则说明 $\boldsymbol{x}_i$ 集成标记的质量较低. 当集成分类器对 $\boldsymbol{x}_i$ 进行纠正时, 采用多数投票 (Majority voting) 的策略对 $\boldsymbol{x}_i$ 进行标注.

2) 当实例 $\boldsymbol{x}_i$ 的集成标记置信度 e_i 小于设定的阈值 T, 则说明 $\boldsymbol{x}_i$ 集成标记的质量较高. 当集成分类器对 $\boldsymbol{x}_i$ 进行纠正时, 采用共识投票 (Consensus voting) 的策略对 $\boldsymbol{x}_i$ 进行标注, 即当三个分类器打标结果相同, 则对 $\boldsymbol{x}_i$ 进行纠正, 否则不纠正 $\boldsymbol{x}_i$ 的标记.

综上所述, 本文提出的 STLNC 算法的详细步骤如算法 1 所示.

算法 1. STLNC 算法

输入. 带集成标记的数据集 $D' = \{(\boldsymbol{x}_i, \hat{y}_i)\}_{i=1}^{N}$, 实例集成标记置信度阈值 T.

输出. 纠正后的干净数据集 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 $i = 1$ to sizeof (D') **do**

根据实例 i 的多噪声标记集 L_i, 获得 i 的标记分布 $\boldsymbol{P}_i$;

end for

for $i = 1$ to sizeof (D') **do**

for $j = 1$ to sizeof (D') && $i != j$ **do**

计算实例 i 和 j 之间的 w_{ij};

根据 w_{ij}, 计算实例 i 和 j 的距离 d_{ij};

end for

根据 d_{ij} 计算实例 i 的局部密度 ρ_i;

根据 d_{ij} 和 ρ_i 计算实例 i 的 δ_i 值;

end for

获取带有指向关系的实例对集合 $D_{\text{couple}} = \{(previous_i,\ next_i)\}_{i=1}^{U}$;

$count = 1$;

while $count > 0$ && sizeof $(D'_n) > 0$ **do**

$count = 0$;

在 D'_c 上训练集成分类器 M;

for $i = 1$ to sizeof (D_{couple}) **do**

if $previous_i \in D'_c$ && $next_i \in D'_n$

通过集成分类器 M 对实例 $next_i$ 按照实例纠正策略进行纠正, 并加入干净集 D'_c;

$count$++;

end if

end for

根据新的干净集和噪声集更新 D_{couple};

end while

$count = 1$;

while $count > 0$ && sizeof $(D'_n) > 0$ **do**

$count = 0$;

在 D'_c 上训练集成分类器 M;

for $i = 1$ to sizeof (D_{couple}) **do**
if $previous_i \in D'_n$ && $next_i \in D'_c$
通过集成分类器 M 对实例 $previous_i$ 按照实例纠正策略进行纠正, 并加入干净集 D'_c;
$count$++;
end if
end for
根据新的干净集和噪声集更新 D_{couple};
end while
获得最后一轮循环训练的集成分类器 M';
用 M' 对 D' 中所有实例进行纠正, 得到 D'_{correct};
return D'_{correct}.

2　实验设计与结果分析

在本节中, 为了验证提出的 STLNC 算法的有效性, 本文在仿真标准数据集和真实众包数据集上进行了实验. 将 STLNC 算法与目前 5 种最先进的众包标记噪声纠正算法 PL[9]、STC[9]、CC[9]、AVNC[11] 和 CENC[13] 在纠正后数据集噪声比以及训练模型的模型质量两个度量指标上进行了比较. 其中, 数据集噪声比 (Noise ratio) 定义为经过算法纠正后的众包数据集中集成标记与真实标记不同的实例所占整个数据集的百分比. 模型质量 (Model quality) 定义为在纠正后的众包数据集上训练的目标分类模型的测试精度.

利用 CEKA (Crowd environment and its knowledge analysis) 平台[17], 本文实现了 STLNC 算法和 CENC 算法, 同时使用了该平台现有的 MV、PL、STC、CC 和 AVNC 算法和 WEKA (Waikato environment for knowledge analysis) 平台[18] 上现有的 K-means 算法、C4.5 算法、KNN (K-nearest neighbor) 算法和 LR (Logistic regression) 算法. 6 种众包标记噪声纠正算法在实验中设置如下:

1) PL: PL 算法使用的基分类器为 C4.5;

2) STC: STC 算法使用的基分类器为 C4.5, 使用的过滤器为 CF (Classification filter) 过滤器[19], 需要被纠正的噪声实例比例设置为 0.8;

3) CC: CC 算法使用的聚类方法为 K-means 聚类算法, 该算法执行次数设置为 10 次, 而簇中心的个数设置为 2 到数据集实例个数的一半;

4) AVNC: AVNC 算法使用的基分类器为 C4.5, 数据集分割为大小相同子集的个数设置为 10 个, 训练的分类器个数设置为 5;

5) CENC: CENC 算法使用的基分类器为 C4.5, 训练的分类器个数设置为 10 个, 而实例的多噪声标记集的信息熵阈值 T 设置为 0.1;

6) STLNC: STLNC 算法使用的集成分类器由 C4.5、KNN ($K = 3$) 和 LR 构成, 使用的过滤器为 CF 过滤器, 截断距离 d_c 中 s 的值设置为 0.2, 实例的多噪声标记集信息熵阈值 T 设置为 0.1.

其中, CF 过滤器将整个数据集分割为 10 个大小相同的子集, 而且所使用的基分类器同样为 C4.5.

2.1　仿真标准数据集上的实验

为了验证 STLNC 在不同工人质量和数据特征下的有效性, 本文使用了 22 个二分类标准数据集由仿真实验模拟不同专业水平的工人进行打标, 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. 为了模拟众包过程为每个实例获取多噪声标记集, 对使用的数据集隐藏了实例的真实标记, 同时模拟了 9 个工人对每个实例打标的过程, 其中每个工人的打标质量为 p_j ($j = 1, 2, \cdots, 9$). 这代表着每个工人有 p_j 的概率给实例打正确标记, $1 - p_j$ 的概率打错误标记. 值得注意的是, 数据集中的正例和负例均有 p_j 概率被正确标记, $1 - p_j$

表 1　22 个仿真标准数据集详细描述
Table 1　Description of 22 simulated benchmark datasets

数据集	#Ins	#Att	#Pos	#Neg
biodeg	1 055	41	356	699
breast-cancer	268	9	85	201
breast-w	699	10	241	458
credit-a	690	16	383	307
credit-g	1 000	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	3 196	37	1 527	1 669
labor	57	16	37	20
mushroom	8 124	23	3 916	4 208
sick	3 772	30	231	3 541
sonar	208	61	111	97
spambase	4 601	57	813	2 788
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	1 268

概率被错误标记.

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 1 个系列的实验中, 所有工人的打标质量设置为 0.6, 即 $p_j = 0.6$;

2) 在第 2 个系列的实验中, 所有工人的打标质量由均匀分布的区间 [0.55, 0.75] 随机生成, 即 $p_j \in [0.55, 0.75]$.

在为每个实例获得 9 个带有噪声的众包标记之后, 本文使用最经典的标记集成算法 MV 获取每个实例的集成标记, 并将该算法处理后的数据集噪声比和训练模型质量作为基线. 然后, 使用 6 种不同的众包标记噪声纠正算法对带有集成标记的众包数据集中的噪声实例进行识别并纠正, 目标分类模型将在纠正后的数据集上进行训练. 最后, 本文将评估每个数据集上各种众包标记噪声纠正算法纠正后的噪声比以及训练模型质量. 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 10 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.

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, 本文采用了威尔科克森 (Wilcoxon) 符号秩检验[20−21] 来比较实验用到的每一对标记噪声纠正算法. 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. 表中, 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 给出了第 1 个系列实验的详细对比, 验证了 STLNC 算法在提高数据集标记质量和模型质量上的有效性. 具体结论如下:

1) STLNC 算法纠正后的数据集的平均噪声比为 12.21%, 低于 MV (27.45%)、PL (21.97%)、STC (19.77%)、CC (18.60%)、AVNC (13.69%) 和 CENC (15.08%).

2) STLNC 算法纠正后的数据集训练的目标模

表 2 工人质量 0.6 时的噪声比对比结果 (%)

Table 2 Noise ratio comparisons with $p_j = 0.6$ (%)

数据集	MV	PL	STC	CC	AVNC	CENC	STLNC
biodeg	28.25	29.95	28.34	19.53	18.48	21.90	15.83
breast-cancer	27.62	26.92	25.87	31.12	26.57	29.37	24.84
breast-w	28.76	9.01	19.31	10.30	9.16	8.44	7.30
credit-a	26.67	20.00	15.94	18.84	13.04	13.33	12.90
credit-g	26.60	27.40	28.40	26.60	25.30	27.50	26.40
diabetes	26.69	32.29	26.56	26.95	23.70	23.96	22.79
heart-statlog	25.19	19.26	23.70	22.96	24.07	25.93	18.52
hepatitis	30.32	19.35	26.45	20.65	27.74	25.16	30.97
horse-colic	27.72	32.34	17.39	21.20	17.66	14.13	14.13
ionosphere	27.92	16.24	21.65	9.12	10.83	13.39	11.68
kr-vs-kp	27.38	21.96	10.45	19.34	2.19	2.85	2.28
labor	31.58	24.56	24.56	15.79	12.28	31.58	7.02
mushroom	26.71	12.65	6.43	4.30	0.04	0.10	0
sick	27.60	2.60	8.83	10.31	1.78	2.28	3.37
sonar	26.92	31.73	29.33	24.04	25.00	24.52	18.75
spambase	27.02	27.47	19.50	14.78	9.11	10.56	8.06
tic-tac-toe	26.20	34.13	23.07	24.43	22.44	22.23	14.61
vote	25.98	4.60	10.34	11.26	3.91	4.14	4.14
climate	27.41	8.52	27.41	14.07	8.52	8.52	8.52
colic	27.45	22.28	20.92	23.10	14.13	14.95	13.59
monks	26.39	25.00	11.34	21.76	5.32	6.71	2.78
steel-plates-faults	27.51	34.98	9.22	18.70	0	0.10	0.15
平均值	27.45	21.97	19.77	18.60	13.69	15.08	**12.21**

表 3 工人质量 0.6 时的模型质量对比结果 (%)
Table 3 Model quality comparisons with $p_j = 0.6$ (%)

数据集	MV	PL	STC	CC	AVNC	CENC	STLNC
biodeg	71.37	75.91	72.13	78.77	74.34	74.21	78.29
breast-cancer	67.00	71.28	69.98	69.08	70.92	68.13	69.73
breast-w	92.85	92.47	90.68	93.38	94.00	92.85	95.54
credit-a	82.03	84.78	84.49	84.78	83.91	83.04	83.48
credit-g	62.20	68.30	67.40	69.70	67.30	63.90	70.40
diabetes	71.74	70.56	71.50	70.80	71.72	72.12	74.00
heart-statlog	65.56	74.81	67.78	70.00	74.07	69.26	78.15
hepatitis	69.00	77.67	71.33	77.50	72.00	70.17	74.17
horse-colic	77.42	78.11	80.00	80.15	81.62	81.36	82.35
ionosphere	83.48	85.45	86.90	85.19	85.77	84.04	85.76
kr-vs-kp	95.18	95.49	96.62	90.29	96.81	96.59	97.03
labor	70.67	61.83	73.50	68.33	68.33	72.33	76.33
mushroom	99.85	98.56	99.88	99.90	99.90	99.86	99.83
sick	96.74	94.62	96.98	94.48	97.77	97.75	97.11
sonar	58.93	55.57	50.07	58.29	55.00	59.14	58.29
spambase	85.92	88.87	87.44	84.20	89.68	88.98	90.39
tic-tac-toe	77.17	69.74	74.54	71.63	74.63	74.63	78.36
vote	89.89	95.37	94.21	90.59	94.21	93.98	94.21
climate	91.48	91.48	91.48	91.48	91.48	91.48	91.48
colic	79.97	81.09	81.09	82.47	81.09	82.47	81.09
monks	90.75	90.73	93.51	83.35	93.51	93.51	93.28
steel-plates-faults	100.00	89.64	100.00	92.01	100.00	100.00	100.00
平均值	80.87	81.47	81.89	81.20	82.64	82.26	**84.06**

表 4 工人质量 0.6 时的噪声比的威尔科克森测试结果
Table 4 Noise ratio summary of Wilcoxon tests with $p_j = 0.6$

	MV	PL	STC	CC	AVNC	CENC	STLNC
MV	—	∘	∘	∘	∘	∘	∘
PL		—		∘	∘	∘	∘
STC	•		—		∘	∘	∘
CC	•			—	∘	∘	∘
AVNC	•	•	•	•	—	•	∘
CENC	•	•	•	•	∘	—	∘
STLNC	•	•	•	•	•	•	—

表 5 工人质量 0.6 时的模型质量的威尔科克森测试结果
Table 5 Model quality summary of Wilcoxon tests with $p_j = 0.6$

	MV	PL	STC	CC	AVNC	CENC	TTLNC
MV	—		∘		∘	∘	∘
PL		—					∘
STC			—		∘		∘
CC				—			∘
AVNC	•		•		—	•	∘
CENC	•					—	∘
STLNC	•	•	•	•	•	•	—

型的平均模型质量为 84.06%, 高于 MV (80.87%)、PL (81.47%)、STC (81.89%)、CC (81.20%)、AVNC (82.64%) 和 CENC (82.26%).

3) 根据威尔科克森符号秩检验的结果, STLNC 算法在噪声比和模型质量两个指标上都要显著优于对比的众包标记噪声纠正算法.

在第 2 个系列实验中, 表 6 ~ 9 详细展示了在工人质量 $p_j \in [0.55, 0.75]$ 的情况下各个众包标记噪声纠正算法在纠正后数据集噪声比和模型质量上的实验结果, 以及威尔科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, 在数据集的噪声比和模型质量两个度量指标上, STLNC 算法的效果都是最好的. 同时, 根据表 8 和表 9 中的威尔科克森符号秩检验的结果, STLNC 算法在纠正后数据集的噪声比上显著优于对比算法 MV、PL、STC、CC, 与 AVNC、CENC 算法性能相当. 而在模型质量上显著优于对比算法 MV、PL、STC、CC、CENC, 与 AVNC 算法性能相当.

根据上述两个系列的实验结果, 在不同的工人质量下的仿真数据集上都验证了本文提出的 STLNC 算法在提高数据集的标记质量和模型质量上的有效性, 并且 STLNC 算法在两个评估指标上都整体优于对比的 5 个众包标记噪声纠正算法.

为了分析实例的多噪声标记集信息熵阈值 T 的取值对 STLNC 算法的影响, 同时节省版面和避免重复, 本文随机选择了仿真数据集“ionosphere”在工人质量 0.6 的情况下进行实验, 同时使用了相同的过滤器 CF 保证每轮实验的干净集和噪声集相同. 最后评估了在不同 T 值的情况下 STLNC 算法在纠正后数据集上噪声比的情况.

在本文中设置 T 值的目的是协助实例纠正策略, 防止集成标记质量高的实例被实例纠正策略给错误纠正. 从图 2 可以看出, 随着 T 值的变化, 噪声比呈现梯度上升, 从而验证了本文设置 T 值的想法, T 值的取值范围为 0.1 ~ 0.5, 为了精确筛选高标记质量实例, 本文设置 T 为较小值 0.1.

2.2 真实众包数据集上的实验

为了进一步验证 STLNC 算法的有效性, 本文使用 CEKA 平台自带的真实众包数据集 Leaves 和

表 6 工人质量 [0.55, 0.75] 时的噪声比对比结果 (%)
Table 6 Noise ratio comparisons with $p_j \in [0.55, 0.75]$ (%)

数据集	MV	PL	STC	CC	AVNC	CENC	STLNC
biodeg	14.22	21.14	16.02	13.84	13.46	13.08	12.89
breast-cancer	16.43	26.22	20.98	19.93	23.43	24.83	24.48
breast-w	20.46	3.72	10.01	4.15	4.15	4.43	3.72
credit-a	18.41	20.58	14.93	13.62	13.77	13.04	12.17
credit-g	17.70	29.60	22.70	22.90	21.60	22.30	24.60
diabetes	20.18	22.66	24.09	22.27	23.44	22.66	23.44
heart-statlog	16.30	20.37	15.19	20.00	16.67	16.67	18.52
hepatitis	12.26	20.65	14.19	14.84	16.77	12.90	12.26
horse-colic	17.66	15.49	13.86	18.75	14.67	14.13	15.22
ionosphere	17.38	18.80	13.68	9.69	11.11	10.83	13.96
kr-vs-kp	17.43	25.19	5.60	11.55	1.31	1.88	2.44
labor	17.54	29.82	17.54	12.28	21.05	21.05	14.04
mushroom	18.07	4.94	4.84	1.67	0.10	0.11	0
sick	13.94	1.78	4.98	3.76	1.46	1.54	2.04
sonar	15.38	37.50	21.63	25.96	19.23	22.60	20.67
spambase	19.32	37.54	15.11	9.04	7.00	7.04	6.67
tic-tac-toe	20.67	27.45	19.31	17.54	15.76	14.41	6.47
vote	22.07	6.90	10.57	8.97	4.37	4.83	4.60
climate	22.96	8.52	22.96	10.74	8.52	8.52	8.52
colic	16.58	19.57	15.49	17.93	15.22	14.40	15.49
monks	17.13	12.73	7.18	23.38	2.78	4.86	2.78
steel-plates-faults	22.46	34.83	7.32	15.92	0.26	0.26	0.21
平均值	17.93	20.27	14.46	14.49	11.64	11.65	**11.15**

表 7 工人质量 [0.55, 0.75] 时的模型质量对比结果 (%)
Table 7 Model quality comparisons with $p_j \in [0.55, 0.75]$ (%)

数据集	MV	PL	STC	CC	AVNC	CENC	STLNC
biodeg	74.58	81.87	76.36	79.99	81.59	80.25	81.78
breast-cancer	69.43	71.81	69.50	70.27	71.28	71.64	71.64
breast-w	90.76	94.54	91.10	94.34	93.85	92.40	94.69
credit-a	82.17	85.36	84.78	85.65	85.65	84.64	84.93
credit-g	69.50	69.80	70.50	69.60	71.10	69.00	72.00
diabetes	71.67	74.44	71.15	74.21	73.13	72.87	74.56
heart-statlog	70.00	78.89	76.30	75.93	79.63	78.52	80.37
hepatitis	76.17	79.17	77.33	80.50	76.83	77.83	79.00
horse-colic	82.50	80.35	81.57	82.63	83.51	83.01	82.68
ionosphere	80.62	83.48	82.33	88.03	87.73	86.90	84.07
kr-vs-kp	97.94	95.34	97.66	92.86	98.28	98.00	98.06
labor	78.33	68.17	78.33	64.33	77.17	77.17	84.33
mushroom	99.99	98.52	99.95	99.96	100.00	100.00	99.95
sick	97.72	96.95	97.48	95.47	97.64	97.83	97.14
sonar	63.79	68.93	67.14	70.36	69.29	69.50	70.86
spambase	86.65	88.87	88.07	84.83	90.37	88.96	90.76
tic-tac-toe	77.83	73.00	77.58	76.30	77.89	76.85	80.08
vote	93.35	93.79	94.98	92.68	95.24	95.00	94.77
climate	91.48	91.48	91.48	91.48	91.48	91.48	91.48
colic	83.74	77.90	84.19	79.17	82.53	82.63	81.84
monks	98.37	85.21	98.60	88.16	100.00	100.00	100.00
steel-plates-faults	99.90	100.00	100.00	99.69	100.00	100.00	100.00
平均值	83.48	83.54	84.38	83.47	85.65	85.20	**86.14**

表 8 工人质量 [0.55, 0.75] 时的噪声比的威尔科克森测试结果
Table 8 Noise ratio summary of Wilcoxon tests with $p_j \in [0.55, 0.75]$

	MV	PL	STC	CC	AVNC	CENC	STLNC
MV	—		∘	∘	∘	∘	∘
PL		—	∘	∘	∘	∘	∘
STC		•	—		∘	∘	∘
CC		•		—	∘	∘	∘
AVNC	•	•	•	•	—		
CENC	•	•	•	•		—	
STLNC	•	•	•	•			—

表 9 工人质量 [0.55, 0.75] 时的模型质量的威尔科克森测试结果
Table 9 Model quality summary of Wilcoxon tests with $p_j \in [0.55, 0.75]$

	MV	PL	STC	CC	AVNC	CENC	STLNC
MV	—		∘		∘	∘	∘
PL		—			∘	∘	∘
STC	•		—		∘	∘	∘
CC				—	∘	∘	∘
AVNC	•	•	•	•	—	•	
CENC	•		•		∘	—	∘
STLNC	•	•	•	•		•	—

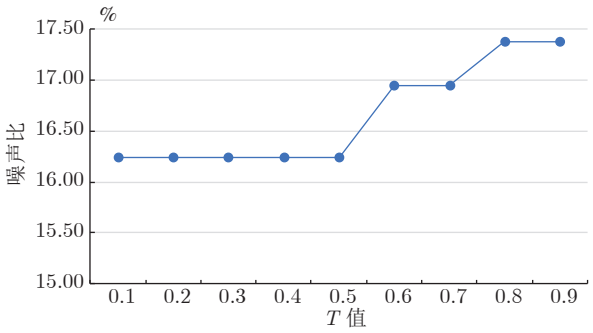


图 2 不同 T 值的 STLNC 在 ionosphere 数据集上的噪声比结果

Fig. 2 Noise ratio of STLNC with different T values on ionosphere dataset

文献 [22] 中使用的真实众包数据集 LabelMe 进行实验. 同时, 在真实众包数据集上的实验, 本文还针对 STLNC 算法过滤阶段过滤器的选取进行了讨论. 最后, 针对 STLNC 中构建空间结构关系阶段和实例选择与纠正阶段进行了消融实验.

Leaves 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图片, 由众包工人根据树叶的特征进行标注, 因此 Leaves 数据集中每个实例都带有多个含噪声的标记. 而 LabelMe 数据集则是包含 1 000 个带有 200 维特征的实例. 每个实例都是一张拍摄的图像, 同样发布在 AMT 平台上, 由不同众包工人进行标注.

本文从 Leaves 和 LabelMe 中提取了 8 个只包含二类的数据集, 这 8 个数据集分别记为 Leaves1, Leaves2, Leaves3, Leaves4, LabelMe1, LabelMe2, LabelMe3 和 LabelMe4, 这些数据集的具体细节如表 10 所示. 例如: "LabelMe1"的目标是判断实例为高速公路还是街道的图片. 该数据集中包含 199 个实例, 其中 89 个为正例 (高速公路), 110 个为负例 (街道). 为获得每个实例的多噪声标记集, 共有 50 个众包工人标注了 395 个标记. 需要注意的是, 在本文的实验中, 针对 LabelMe 数据集, 首先使用文献 [23] 中图像处理的方法将图像数据转换为文本词向量, 然后通过文本分类器对数据集进行训练和预测. 其中, 使用 MNB (Multinomial naive Bayes) 算法[24] 作为 PL、STC、CC、AVNC 和 CENC 的基分类器, 而 STLNC 的基分类器则替换为 MNB、CNB (Complement naive Bayes)[25] 和 OVA (One-versus-all-but-one)[25].

图 3 ~ 6 分别展示了各个算法在 8 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: 在 Leaves 数据集上, STLNC 算法的效果整体优于 MV、PL、STC、CC、AVNC 以及 CENC 算法, 而在 LabelMe 数据集上, STLNC 算法的优势更加显著. 因此, 通过上述实验的比较, STLNC 同时提高了数据集的标记质量和模型质量, 验证了该算法在真实众包场景的有效性.

为了分析过滤器的选取对 STLNC 算法的影响, 本文选取了传统机器学习领域经典的噪声过滤器 CF、IPF (Iterative partitioning filtering)[26] 和 MVF (Majority vote filter)[27], 三个过滤器具体描述如下:

1) 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, 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 MNB.

2) IPF: IPF 的思想将整个数据集分为大小相等的 n 个子集, 在每个子集上训练一个分类器来对数据集中每个实例进行预测, 再根据投票策略判断实例是否为噪声, 如果是则移出数据集. 执行上述的步骤直到筛选出的噪声实例的比例小于阈值时停止. 在本实验中 IPF 的 n 为 5, 阈值为 0.01, 投票策

表 10 8 个真实众包数据的详细描述

Table 10 Description of eight real-world crowdsourced datasets

数据集	分类任务	#Instances	#Positives	#Negatives	#Labelers	#Labels
Leaves1	maple/alder	142	96	46	70	1 093
Leaves2	maple/tilia	140	96	44	74	1 044
Leaves3	alder/eucalyptus	93	46	47	58	407
Leaves4	alder/poplar	89	46	43	54	400
LabelMe1	highway/street	199	89	110	50	395
LabelMe2	highway/forest	227	89	138	54	476
LabelMe3	highway/opencountry	240	89	151	54	375
LabelMe4	highway/insidecity	205	89	116	49	339

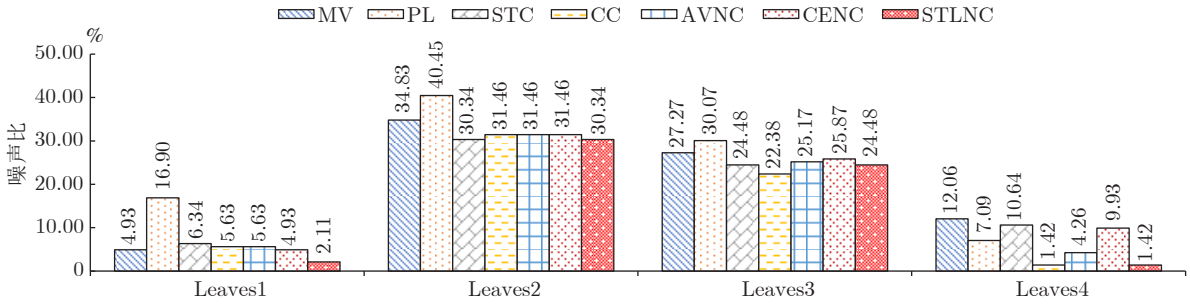


图 3　Leaves 数据集上的噪声比对比结果

Fig. 3　Noise ratio comparisons on Leaves datasets

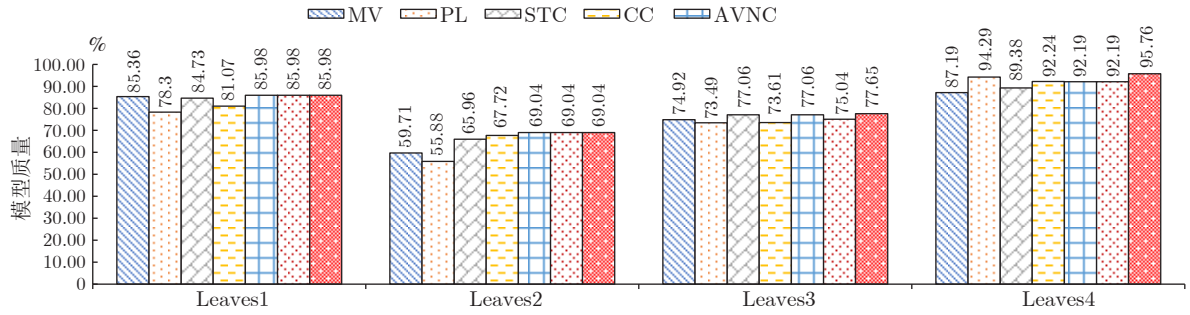


图 4　Leaves 数据集上的模型质量结果对比结果

Fig. 4　Model quality comparisons on Leaves datasets

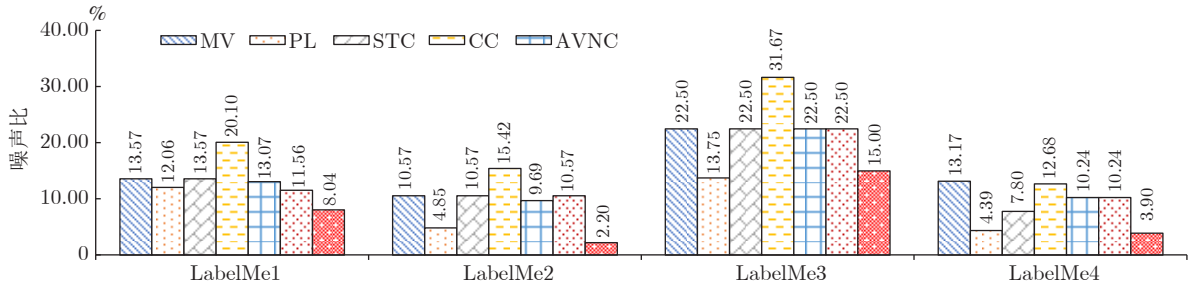


图 5　LabelMe 数据集上的噪声比对比结果

Fig. 5　Noise ratio comparisons on LabelMe datasets

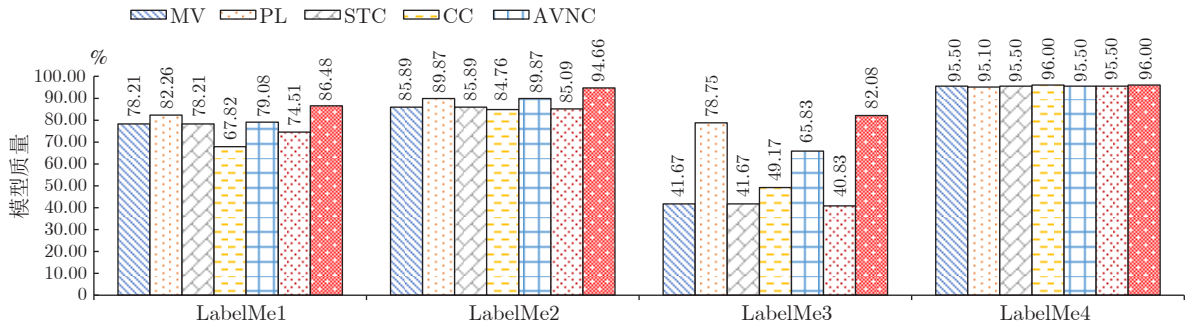


图 6　LabelMe 数据集上的模型质量对比结果

Fig. 6　Model quality comparisons on LabelMe datasets

略为多数投票, 所使用的基分类器为 MNB.

3) 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 MNB、CNB 和 OVA.

根据以上噪声过滤器设置, 本文随机选取了真实众包数据集 LabelMe4 进行了实验, 并评估了使用不同过滤器的 STLNC 在纠正后的数据集上的噪声比和训练模型质量.

图 7(a) 和图 7(b) 分别展示了使用各个过滤器的 STLNC 算法在 LabelMe4 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: 使用 CF、IPF 和 MVF 过滤器的 STLNC 在噪声比和模型质量上结果相似. 由此分析可得, STLNC 在选用不同过滤器时表现稳定, 对算法性能影响不大.

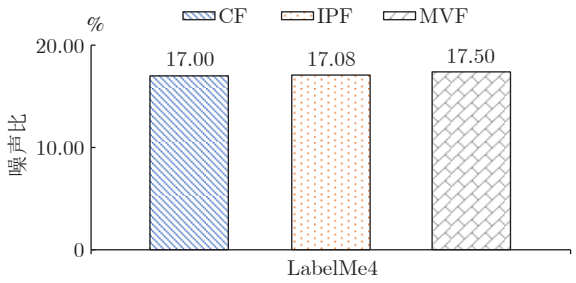
在第 2 节中, 本文介绍了在构建空间结构关系阶段和实例选择与纠正阶段的主要改进为: 1) 使用众包数据信息对密度峰值聚类进行加权, 构建空间结构关系; 2) 使用集成分类器, 同时设计实例选择和纠正策略. 为了详细说明 STLNC 算法的这两个核心阶段的重要性, 本文还在 LabelMe4 上进行了消融实验. 消融实验对比算法说明如下:

1) STLNC-(1): 表示该算法与 STLNC 相比, 未使用加权的密度峰值聚类构建空间结构关系, 只使用原始的密度峰值聚类算法.

2) STLNC-(2): 表示该算法与 STLNC 相比, 未使用集成分类器和实例选择与纠正策略. 只使用单个分类器 MNB 对噪声实例进行预测, 将预测后的标记就作为纠正的结果.

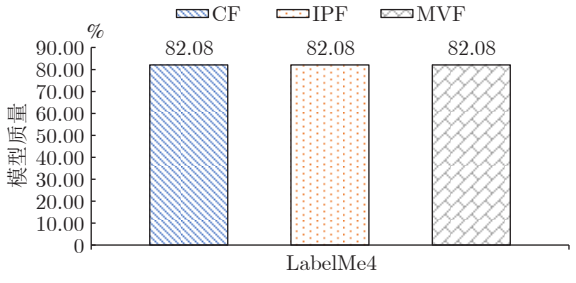
3) STLNC-(1)(2): 表示该算法与 STLNC 相比, 未使用加权的密度峰值聚类构建空间结构关系以及未使用集成分类器和实例选择与纠正策略. 只使用原始的密度峰值聚类算法构建空间结构关系, 使用了单个分类器 MNB 对噪声实例进行预测, 将预测后的标记就作为纠正的结果.

从图 8(a) 和图 8(b) 中可以看出, STLNC 算法在纠正后数据集上的噪声比和训练的模型质量表现最好, STLNC-(1) 和 STLNC-(2) 算法表现其次, STLNC-(1)(2) 的算法效果最差. 从而可以得出结论: STLNC 中两个核心阶段的改进均能提高 STLNC 算法的性能.



(a) 噪声比实验结果

(a) Experimental results of noise ratio

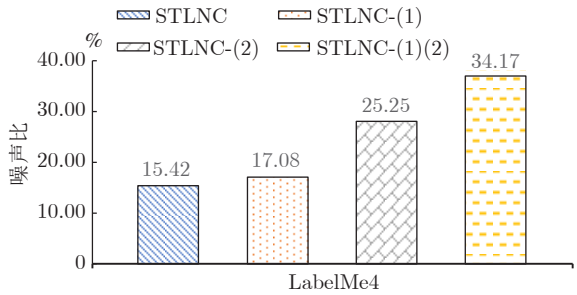


(b) 模型质量实验结果

(b) Experimental results of model quality

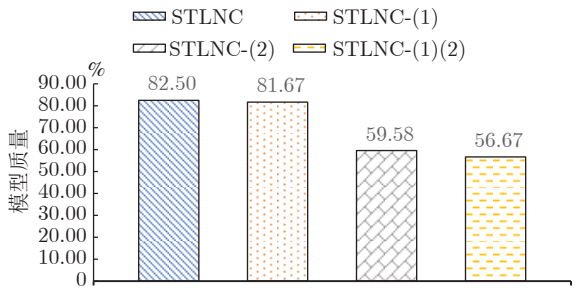
图 7 STLNC 基于不同过滤器在 LabelMe4 数据集上的实验结果

Fig. 7 Experimental results of STLNC with different filters on LabelMe4 dataset



(a) 噪声比实验结果

(a) Experimental results of noise ratio



(b) 模型质量实验结果

(b) Experimental results of model quality

图 8 STLNC 在 LabelMe4 数据集上的消融实验结果

Fig. 8 Results of STLNC ablation experiment on LabelMe4 dataset

3 结束语

本文针对众包数据集经过标记集成后仍然存在标记噪声问题, 提出了一种基于自训练的众包标记噪声纠正算法 STLNC. 主要创新包括: 1) 在构建空间结构关系阶段, 本文将实例的多噪声标记集转化为标记分布, 根据标记分布信息改进了密度峰值聚类算法, 从而精确地构建数据集中低密度实例指向高密度实例的空间结构关系; 2) 在实例选择与纠正阶段, 根据构建的空间结构关系, 设计了噪声实例选择策略, 同时引入集成学习的思想和标记置信度计算, 设计了新的实例纠正策略, 提高了纠正的准确性.

根据在 22 个仿真的标准数据集以及 8 个真实的众包数据集上的实验结果, STLNC 与 PL、STC、CC、AVNC、CENC 五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众包标记噪声纠正算法的有效性和优越性. 但是, 本文方法在算法的过滤阶段并未进行深入的研究, 如何获取更加精确的干净集和噪声集仍然是未来将要面临的难题之一. 因此, 下一步工作将围绕设计更好的过滤策略, 提高过滤效果展开.

References

1 Pollicelli D, Coscarella M, Delrieux C. RoI detection and segmentation algorithms for marine mammals photo-identification. *Ecological Informatics*, 2020, **56**: Article No. 101038

2 Wang H, Zhao D, Ma H D. Informative image selection for crowdsourcing-based mobile location recognition. *Multimedia Systems*, 2019, **25**(5): 513−523

3 Lotfian R, Busso C. Building naturalistic emotionally balanced speech corpus by retrieving emotional speech from existing podcast recordings. *IEEE Transactions on Affective Computing*, 2019, **10**(4): 471−483

4 Sheng V S, Provost F, Ipeirotis P G. Get another label? Improving data quality and data mining using multiple, noisy labelers. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Las Vegas, USA: ACM, 2008. 614−622

5 Demartini G, Difallah D E, Cudré-Mauroux P. ZenCrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In: Proceedings of the 21st International Conference on World Wide Web. Lyon, France: ACM, 2012. 469−478

6 Zhang H, Jiang L Z, Xu W Q. Multiple noisy label distribution propagation for crowdsourcing. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence. Macao, China: AAAI Press, 2019. 1473−1479

7 Tian T, Zhu J, You Q B. Max-margin majority voting for learning from crowds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019, **41**(10): 2480−2494

8 Zhong J H, Yang P, Tang K. A quality-sensitive method for learning from crowds. *IEEE Transactions on Knowledge and Data Engineering*, 2017, **29**(12): 2643−2654

9 Nicholson B, Sheng V S, Zhang J. Label noise correction and application in crowdsourcing. *Expert Systems With Applications*, 2016, **66**: 149−162

10 Xu W Q, Jiang L X, Li C Q. Resampling-based noise correction for crowdsourcing. *Journal of Experimental and Theoretical Artificial Intelligence*, 2021, **33**(6): 985−999

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

12 Li C Q, Jiang L X, Xu W Q. Noise correction to improve data and model quality for crowdsourcing. *Engineering Applications of Artificial Intelligence*, 2019, **82**: 184−191

13 Xu W Q, Jiang L X, Li C Q. Improving data and model quality in crowdsourcing using cross-entropy-based noise correction. *Information Sciences*, 2021, **546**: 803−814

14 Wu D, Shang M S, Luo X, Xu J, Yan H Y, Deng W H, et al. Self-training semi-supervised classification based on density peaks of data. *Neurocomputing*, 2018, **275**: 180−191

15 Khuri S A, Sayfy A. A laplace variational iteration strategy for the solution of differential equations. *Applied Mathematics Letters*, 2012, **25**(12): 2298−2305

16 Hershey J R, Olsen P A. Approximating the kullback leibler divergence between Gaussian mixture models. In: Proceedings of the 32nd IEEE International Conference on Acoustics, Speech and Signal Processing. Honolulu, USA: IEEE, 2007. IV-317−IV-320

17 Zhang J, Sheng V S, Nicholson B, Wu X D. CEKA: A tool for mining the wisdom of crowds. *The Journal of Machine Learning Research*, 2015, **16**(1): 2853−2858

18 Witten I H, Frank E. *Data Mining: Practical Machine Learning Tools and Techniques*. (3rd edition). Beijing: China Machine Press, 2005.

19 Gamberger D, Lavrac N, Groselj C. Experiments with noise filtering in a medical domain. In: Proceedings of the 16th International Conference on Machine Learning. Bled, Slovenia: ACM, 1999. 143−151

20 García S, Herrera F. An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research*, 2008, **9**(12): 2677−2694

21 Jiang L X, Zhang L G, Li C Q, Wu J. A correlation-based feature weighting filter for naive Bayes. *IEEE Transactions on Knowledge and Data Engineering*, 2019, **31**(2): 201−213

22 Rodrigues F, Lourenčo M, Ribeiro B, Pereira F C. Learning supervised topic models for classification and regression from crowds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, **39**(12): 2409−2422

23 Li F F, Perona P. A Bayesian hierarchical model for learning natural scene categories. In: Proceedings of the 14th IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Diego, USA: IEEE, 2005. 524−531

24 McCallum A, Nigam K. A comparison of event models for naive Bayes text classification. In: Proceedings of the AAAI-98 Workshop on Learning for Text Categorization. Palo Alto, USA: AAAI Press, 1998. 41−48

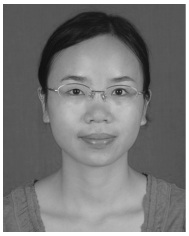
25 Rennie J D M, Shih L, Teevan J, Karger D R. Tackling the poor assumptions of naive Bayes text classifiers. In: Proceedings of the 20th International Conference on Machine Learning. Washington, USA: AAAI Press, 2003. 616−623

26 Khoshgoftaar T M, Rebours P. Improving software quality prediction by noise filtering techniques. *Journal of Computer Science and Technology*, 2007, **22**(3): 387−396

27 Brodley C E, Friedl M A. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 1999, **11**(1): 131−167

杨 艺 中国地质大学 (武汉) 计算机学院硕士研究生. 2018 年获得中国地质大学 (武汉) 计算机学院学士学位. 主要研究方向为机器学习与数据挖掘. E-mail: yangyi@cug.edu.cn
(**YANG Yi** Master student at the School of Computer Science, China University of Geosciences (Wuhan). He received his bachelor degree from China University of Geosciences (Wuhan) in 2018. His research interest covers machine learning and data mining.)

蒋良孝 中国地质大学 (武汉) 计算机学院教授. 2009 年获得中国地质大学 (武汉) 地球探测与信息技术博士学位. 主要研究方向为机器学习与数据挖掘. 本文通信作者.
E-mail: ljiang@cug.edu.cn
(**JIANG Liang-Xiao** Professor at the School of Computer Science, China University of Geosciences (Wuhan). He received his Ph.D. degree in earth prospecting and information technology from China University of Geosciences (Wuhan) in 2009 . His research interest covers machine learning and data mining. Corresponding author of this paper.)



李超群 中国地质大学 (武汉) 数学与物理学院副教授. 2012 年获得中国地质大学 (武汉) 地球探测与信息技术博士学位. 主要研究方向为机器学习与数据挖掘.
E-mail: chqli@cug.edu.cn
(**LI Chao-Qun** Associate professor at the School of Mathematics and Physics, China University of Geosciences (Wuhan). She received her Ph.D. degree in earth prospecting and information technology from China University of Geosciences (Wuhan) in 2012. Her research interest covers machine learning and data mining.)