DOI: 10.11992/tis.201804052
网络出版地址: http://kns.cnki.net/kcms/detail/23.1538.TP.20180627.1529.002.html

基于 PageRank 的主动学习算法

邓思宇[1], 刘福伦[1], 黄雨婷[1], 汪敏[2]
（1. 西南石油大学 计算机科学学院, 四川 成都 610500; 2. 西南石油大学 电气信息学院, 四川 成都 610500）

摘　要: 在许多分类任务中, 存在大量未标记的样本, 并且获取样本标签耗时且昂贵。利用主动学习算法确定最应被标记的关键样本, 来构建高精度分类器, 可以最大限度地减少标记成本。本文提出一种基于 PageRank 的主动学习算法 (PAL), 充分利用数据分布信息进行有效的样本选择。利用 PageRank 根据样本间的相似度关系依次计算邻域、分值矩阵和排名向量; 选择代表样本, 并根据其相似度关系构建二叉树, 利用该二叉树对代表样本进行聚类, 标记和预测; 将代表样本作为训练集, 对其他样本进行分类。实验采用 8 个公开数据集, 与 5 种传统的分类算法和 3 种流行的主动学习算法比较, 结果表明 PAL 算法能取得更好的分类效果。

关键词: 分类; 主动学习; PageRank; 邻域; 聚类; 二叉树
中图分类号: TP181　　**文献标志码:** A　　**文章编号:** 1673−4785(2019)03−0551−09

中文引用格式: 邓思宇, 刘福伦, 黄雨婷, 等. 基于 PageRank 的主动学习算法[J]. 智能系统学报, 2019, 14(3): 551–559.
英文引用格式: DENG Siyu, LIU Fulun, HUANG Yuting, et al. Active learning through PageRank[J]. CAAI transactions on intelligent systems, 2019, 14(3): 551–559.

Active learning through PageRank

DENG Siyu[1], LIU Fulun[1], HUANG Yuting[1], WANG Min[2]
(1. School of Computer Science, Southwest Petroleum University, Chengdu 610500, China; 2. School of Electrical Engineering and Information, Southwest Petroleum University, Chengdu 610500, China)

Abstract: In many classification tasks, there are a large number of unlabeled samples, and it is expensive and time-consuming to obtain a label for each class. The goal of active learning is to train an accurate classifier with minimum cost by labeling the most informative samples. In this paper, we propose a PageRank-based active learning algorithm (PAL), which makes full use of sample distribution information for effective sample selection. First, based on the PageRank theory, we sequentially calculate the neighborhoods, score matrices, and ranking vectors based on similarity relationships in the data. Next, we select representative samples and establish a binary tree to express the relationships between representative samples. Then, we use a binary tree to cluster, label, and predict representative samples. Lastly, we regard the representative samples as training sets for classifying other samples. We conducted experiments on eight datasets to compare the performance of our proposed algorithm with those of five traditional classification algorithms and three state-of-the-art active learning algorithms. The results demonstrate that PAL obtained higher classification accuracy.
Keywords: classification; active learning; PageRank; neighborhood; clustering; binary tree

传统的监督学习算法, 如 Naïve Bayes[1]、One-R[2]和 J48[3]等, 其分类效果依赖于训练数据的有效性。通常情况下, 使用已标记的样本作为训练集, 学习算法以此训练出分类模型。然而, 在真实的数据分析场景下, 大量的无标注样本较易获取, 而已标注样本数量稀少且难以获取。对海量数据进行标注是耗时、昂贵且困难的。在此情况下, 半监督学习 (semi-supervised learning)[4]和主动学习 (active learning)[5]被提出并得到快速发展, 已经被广泛地应用在文本分类[6]、语音识别[7]和图像分类[8]等领域。

收稿日期: 2018−04−26.　**网络出版日期:** 2018−06−28.
基金项目: 国家自然科学基金项目 (61379089).
通信作者: 汪敏. E-mail: wangmin80616@163.com.

主动学习模拟一种人机交互场景，允许学习算法根据查询策略，主动获取选取样本的真实类标签，对主动标注的样本进行训练，不断修正已有分类模型，从而提高分类器的泛化能力和分类精度。因此，主动学习的主要挑战是制定有效的样本选择策略。目前，比较常见的主动学习方法有不确定抽样 (sampling uncertainty, UC)[9]，基于聚类 (clustering-based approaches, CBA)[10]和基于委员会投票采样法 (query-by-committee, QBC)[11]。其中，不确定性抽样方法选择当前分类器中不确定度最高的未标注样本进行标注，并将其添加到训练集中。由于单一分类器存在分类偏好，使得泛化能力产生定式，而 QBC 通过多种同质或异质分类器共同参与分类，一般选取冲突性 (不一致性) 最高的未标注样本进行标注。基于聚类的样本选择方法旨在通过分析样本间的内在相似性，对样本进行划簇，而后从每簇中选择代表样本进行标注。

PageRank[12]建立在随机冲浪模型上，通过计算网页的 PageRank 分值，解决了互联网搜索引擎的网页排名问题。PageRank 理论基于两个简单的假设：1) 较重要的网页被更多的网页链接；2)PageRank 分值越高的网页将传递更高的权重。本文结合 PageRank 理论，将 PageRank 分值作为样本信息量的度量指标，同时充分考虑样本的分布信息，提出一种基于 PageRank 的主动学习算法 (PageRank-based active learning algorithm, PAL)，为主动学习算法中样本的选择问题提供一种可行的方案。

实验在 8 个公开数据集上进行，通过设置不同规模的训练集，测试 PAL 算法的分类性能。实验结果表明，PAL 算法较 Naïve Bayes、J48、kNN[13]和 One-R 等经典分类算法，通常能得到更高的分类精度，且与 QBC、KQBC[14]和 MADE[15]等主动学习算法相比，有更好的分类性能。

1　数据模型

在本节中，主要介绍决策信息系统、PageRank 理论等基本概念。

1.1　决策信息系统

定义 1　决策信息系统[16]。决策信息系统定义成一个三元组：

$$S = (U, C, d) \tag{1}$$

式中：U 代表一个非空样本集合，也称论域；C 代表一个非空条件属性集合；d 指的是样本的决策属性。表 1 是 1 个决策信息系统，$U = \{x_0, x_1, x_2, \cdots, x_{15}\}$，$C = \{a_1, a_2, a_3, a_4\}$。

表 1　决策信息系统

Table 1　Example of decision system

U	a_1	a_2	a_3	a_4	d
x_0	5.1	3.5	1.4	0.2	is
x_1	4.9	3.0	1.4	0.2	is
x_2	4.7	3.2	1.3	0.2	is
x_3	4.6	3.1	1.5	0.2	is
x_4	5.0	3.6	1.4	0.2	is
x_5	5.4	3.9	1.7	0.4	is
x_6	7.0	3.2	4.7	1.4	iv
x_7	6.4	3.2	4.5	1.5	iv
x_8	6.9	3.1	4.9	1.5	iv
x_9	5.5	2.3	4.0	1.3	iv
x_{10}	6.5	2.8	4.6	1.5	iv
x_{11}	6.3	3.3	6.0	2.5	it
x_{12}	5.8	2.7	5.1	1.9	it
x_{13}	7.1	3.0	5.9	2.1	it
x_{14}	6.5	3.0	5.8	2.2	it
x_{15}	7.6	3.0	6.6	2.1	it

定义 2　曼哈顿距离。向量 $\boldsymbol{x} = [a_1\ a_2 \cdots\ a_m]$ 与 $\boldsymbol{y} = [b_1\ b_2 \cdots\ b_m]$ 的曼哈顿距离为

$$\mathrm{dis}(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i=1}^{m} |a_i - b_i| \tag{2}$$

式 (2) 表示在多维空间中两个点之间的距离。信息表的样本可以用向量表示。相应地，可以定义任意一组样本的相似度。

定义 3　相似度。给定一个决策信息系统 $S = (U, C, d)$，任意 $x, y \in U$ 的相似度记为

$$\mathrm{sim}(x, y) = \frac{1}{1 + \mathrm{dis}(x, y)} \tag{3}$$

根据式 (2)、式 (3)，可计算表 1 的决策信息系统中 $\mathrm{sim}(x_0, x_6) = 0.13$，$\mathrm{sim}(x_3, x_{12}) = 0.127$。

定义 4　邻域。对于任意的样本 $x \in U$，可以通过设置相似度阈值 θ 的方式确定其邻域，样本的邻域定义为

$$n(x, \theta) = \{y \in U | \mathrm{sim}(x, y) \geqslant \theta\} \tag{4}$$

相似度阈值 θ 越小，样本的邻域越大。根据表 1 所示的决策信息系统可以计算出 $n(x_0, 0.5) = \{x_1, x_2, x_3, x_4\}$。

1.2　PageRank 模型

Web 中的网页通过超链接相互链接，PageRank 算法计算每个网页的 PageRank 分值。Page-

Rank 分值可作为网页重要程度的度量指标。图 1 表示一个 Web 超链接图。

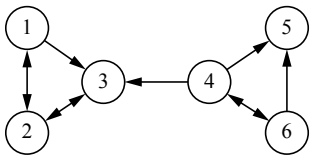


图 1　超链接网络
Fig. 1　Hyperlink network

定义 5　PageRank 分值。将互联网中的网页抽象成一个有向图 $G=(V,E)$。E 是网页超链接集合, V 是网页集合。设 $n=|V|$, 网页 i 的 PageRank 分值 point(i) 定义为

$$\text{point}(i)=\sum_{(j,i)\in E}\frac{\text{point}(j)}{O_j} \tag{5}$$

式中 O_j 表示网页 j 的出度。此时, PageRank 分值的 n 维行向量可用 $\boldsymbol{P}$ 表示, 即

$$\boldsymbol{P}=[\text{point}(1)\ \ \text{point}(2)\ \ \cdots\ \ \text{point}(n)]^{\mathrm{T}} \tag{6}$$

有向图 G 的邻接矩阵可以用 $\boldsymbol{A}=(a_{ij})_{n\times n}$ 表示, 其中:

$$a_{ij}=\begin{cases}1/O_i, & (i,j)\in E\\ 0, & \text{其他}\end{cases} \tag{7}$$

根据式 (6)、式 (7) 可定义 n 维方程组为

$$\boldsymbol{P}_i=\boldsymbol{A}^{\mathrm{T}}\boldsymbol{P}_{i-1} \tag{8}$$

式 (8) 是循环定义式。迭代求得分值向量 $\boldsymbol{P}$, 即 $\boldsymbol{P}$ 不再显著变化或者趋近收敛时, 停止迭代。初始情况下, 所有网页的排名是相同的, 即 $\boldsymbol{P}_0=[1\ 1\cdots 1]$。极小值 ε 是人工设定的收敛阈值, 用于验证向量 $\boldsymbol{P}$ 是否收敛。每轮迭代结束后, 若 $|\boldsymbol{P}_i-\boldsymbol{P}_{i-1}|<\varepsilon$, 则认为达到收敛条件。

在有向图 G 中, 存在没有出度的网页 v, 称之为悬挂网页, 如图 1 中的 V_5。悬挂网页导致排名下沉, PageRank 分值向量 $\boldsymbol{P}$ 在经过 i 次迭代后, 其值均为 0。将 Web 图用马尔可夫链[17]进行建模可以解决上述问题。

将网页看作马尔可夫链的状态, 超链接表示状态转移。这样, Web 冲浪将表示成一种随机过程。状态转移矩阵 $\boldsymbol{T}$ 必须满足 3 个条件: 随机矩阵、不可约、非周期。因此, 将邻接矩阵 $\boldsymbol{A}$ 进行如下修订:

$$\boldsymbol{T}=\gamma\boldsymbol{A}^{\mathrm{T}}+(1-\gamma)\frac{\boldsymbol{E}}{n} \tag{9}$$

式中: γ 是阻尼系数, 一般情况下 $\gamma\in(0,1)$; $\boldsymbol{E}$ 是一个 $n\times n$ 阶且元素全为 1 的矩阵; $\boldsymbol{E}/n$ 表示一网页链接其他网页的随机概率, 即 $1/n$。

2　问题与算法

2.1　问题描述

在主动学习应用场景中, 算法标注最具信息量的样本来构建高精度分类器。可供查询的标签数量 N 是输入参数之一。

输入　决策信息系统 $S=(U,C,d)$;
输出　分类精度 accuracy;
约束条件　$U=U_r\cup U_t$。
优化目标　最大化精度 accuracy, 即

$$\text{accuracy}=\frac{|U_t|-\text{error}}{|U_t|}\times 100\% \tag{10}$$

式中: $|U_r|$是训练集大小; $|U_t|$是测试集大小; error 是误分类样本数量。若可供查询的标签数量为 N, 则 $|U_t|=n-N$。

2.2　PAL 算法描述

PAL 算法可以细分为 3 个子算法, 分别是 PageRank 排名计算算法、二叉树生成算法和二叉树聚类算法。伪代码符号定义如表 2。

表 2　符号定义
Table 2　Symbol definitions

符号	定义说明
U	所有样本集合
U'	排名前 R 的样本集合
$\boldsymbol{T}$	状态转移矩阵
$\boldsymbol{P}$	分值向量
Rank	排名向量
x_l	左孩子样本
x_r	右孩子样本
U_l	集合 $U_l\subseteq U'$ 为 x_l 的样本集合
U_r	集合 $U_r\subseteq U'$ 为 x_r 的样本集合
setChild ()	设置子节点 (函数)
cluster$\boldsymbol{T}$ ()	聚类 (函数)
cn	记录簇号的数组
bl_i	第 i 簇信息块

2.2.1　PageRank 排名计算算法

利用 PageRank 计算每个样本的分值, 该分值可作为样本信息量的度量标准, 即分值越大样本所含信息量越高。

给定决策信息类系统 $S=(U,C,d)$, 对于任意的 $x,x'\in U$, 且 $x\in n(x',\theta)$。根据式 (4)、式 (5) 计算样本 x 在 PageRank 模型下所获得的分数 point(x):

$$\text{point}(x)=\sum_{x\in(x',\theta)}\frac{\text{point}(x')}{|n(x',\theta)|} \tag{11}$$

根据式 (7) 决策系统邻接矩阵可用 $\boldsymbol{A}'=(a_{ij}')_{n\times n}$ 表示, 其中:

$$a'_{ij}=\begin{cases}1/|n(x_i,\theta)|, & x_j\in n(x_i,\theta)\\ 0, & \text{其他}\end{cases} \tag{12}$$

算法 1 PageRank 排名计算算法

输入 决策信息系统 $S=(U, C, d)$;

输出 排名向量 **Rank**。

1) for (each $x\in U$) do

2) for (each $y\in U$) do

3) 根据式 (2) 计算 dis(x y);

4) 根据式 (3) 计算 sim(x, y);

5) end for

6) 根据式 (4) 计算 $n(x)$;

7) end for

8) 根据式 (12) 计算邻接矩阵 $\boldsymbol{A}$;

9) $k=1$;

10) $\boldsymbol{T}=\gamma\boldsymbol{A}^{\mathrm{T}}+(1-\gamma)E/n$;

11) $\boldsymbol{P}_0=[11\cdots1]^{\mathrm{T}}_{1\times n}$;

12) while ($|\boldsymbol{P}_k-\boldsymbol{P}_{k\text{-}1}|>\varepsilon$) do

13) $\boldsymbol{P}_k=\boldsymbol{T}\boldsymbol{P}_{k\text{-}1}$;

14) k++;

15) end while

16) Rank = sort($\boldsymbol{P}_k$);

17) return Rank;

算法 1 描述了样本的排名向量的计算过程。1)~7) 通过计算样本间的相似度，确定每个样本的邻域；10) 根据式 (9) 计算状态转移矩阵 $\boldsymbol{T}$；11) 定义初始分值向量 $\boldsymbol{P}_0$；12)~15) 计算收敛条件下分值矩阵 $\boldsymbol{P}$；16) 对分值矩阵 $\boldsymbol{P}$ 进行降序排序。

2.2.2 二叉树生成算法

主动学习阶段在二叉树上进行，为了避免离群点对该阶段标签查询、预测的影响，保证查询到的样本均具有较高的信息量，仅利用排名前 R 的代表样本去构建二叉树。同时，树形结构能够充分体现数据的层次关系，便于数据分析，从而得到更好的聚类结果。

二叉树生成算法是一个典型递归算法。其构建过程分为两步：寻找孩子节点，根据孩子节点划分集合。根结点 root 的孩子节点是 U'中最不相似的两个样本，其余节点 x 的左孩子是当前集合中与 x 最相似的样本 x_l，右节点是当前集合中与 x_l 最不相似的节点 x_r。

算法 2 二叉树生成算法

输入 排名前 R 的代表集合 U';

输出 二叉树 root。

1) while (U' != Ø) then

2) if (root) then

3) 计算最不相似的一对样本 $x_l, x_r\in U'$;

4) root. setChild (x_l, x_r);

5) $U'=U'-\{x_l, x_r\}$;

6) else then

7) 计算与 root 最相似的样本 $x_l\in U'$;

8) 计算与 x_l 最不相似的样本 $x_r\in U'$;

9) root. setChild (x_l, x_r);

10) $U'=U'-\{x_l, x_r\}$;

11) end if

12) $U_l=$ Ø, $U_r=$ Ø;

13) for (each $x\in U'$) do

14) if (sim(x, x_l) > sim(x, x_r)) then

15) $U_l=U_l\cup\{x\}$;

16) else then

17) $U_r=U_r\cup\{x\}$;

18) end if

19) end for

20) 对 x_l, U_l 继续调用算法 2;

21) 对 x_r, U_r 继续调用算法 2;

22) end while

23) return root;

3)~5) 寻找 root 的孩子节点，即 U'中最不相似的一对样本；7)~10) 寻找非 root 节点的孩子节点；12) 定义 x_l 和 x_r 的样本集合；13)~19) 通过比较集合 U'中样本与 x_l、x 相似度大小，实现集合的划分；20)~21) 递归调用算法 2。

2.2.3 二叉树聚类算法

一般来说，聚类簇数 K 与聚类质量关系密切，然而大多数聚类算法只能通过经验或者试凑指定簇数 K。本文采用一种执行边缘分离的聚类策略，不需要将 K 作为输入，而是根据二叉树的内部结构自然地分簇。

通过计算二叉树节点间的相似度，将二叉树的边划分为分割边或者非分割边。假设两节点足够相似，可将该连边定义成非分割边，反之定义为分割边。这种边界划分方式基于一个阈值。第一轮迭代时，阈值是二叉树相连节点间相似度的最小值。

算法 3 二叉树聚类算法

输入 二叉树根节点 root;

输出 聚类信息块 **bl** = [bl_1 $\mathrm{bl}_2\cdots\mathrm{bl}_k$]。

1) cluster$\boldsymbol{T}$ (root.lc, count, threshold) start

2) cn[root.lc] = count;

3) if (root.lc.lc! = null) then

4) if (sim (root.lc, root.lc.lc) ≤ threshold)

5) cluster$\boldsymbol{T}$ (root.lc.lc, ++count, threshold);

6) else then

7) cluster***T*** (root.lc.lc, count, threshold);
8) end if
9) end if
10) if (root.lc.rc! = null) then
11) 步骤 4)、5)、6)、7);
12) end if
13) end cluster***T*** (root.lc, count, threshold)
14) for(i = 0 to count) do
15) tempNum= 0;
16) for(j = 0 to N) do
17) if(cn_j = i)then
18) bl(i, tempNum) = j; tempNum ++;
19) end if;
20) end for
21) end for
22) return bl;

算法 3 详细描述了基于二叉树的聚类过程。通过遍历树的节点, 同时用数组 cn 记录节点的簇号, 实现聚类。lc 表示左孩子, 同理 rc 表示右孩子。count 用于记录递归过程中最大簇数。1) 定义聚类函数; 2) 记录节点的簇号; 3)~9) 根据相似度关系判断簇边界, 如当前节点与它的孩子节点的相似度小于阈值 threshold, count 自增后进行下一次递归; 14)~21) 整理 cn 得到分块信息表 bl。该方法可以解决聚类算法需要人工设定 K 值的问题。

2.2.4　主动学习

主动学习阶段, 利用二叉树聚类算法生成的信息块 bl 对代表样本进行标记和预测。

1) 如 bl_i 中存在未分类样本, 则查询 bl_i 中 PageRank 值较高的一部分样本的标签。

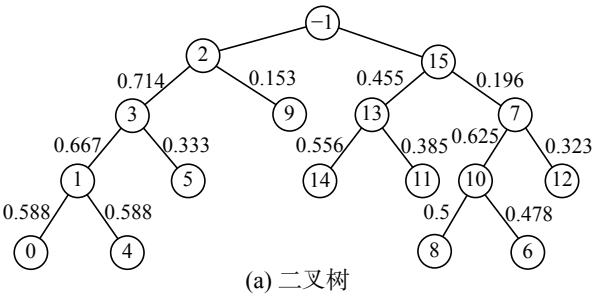
2) 如 bl_i 中已分类的样本数量足够大 ($P_i \geqslant \sqrt{bl_i}$) 且标签一致, 则可预测该块中剩余样本的标签。

3) 增大阈值 threshold, 进行下一轮聚类、标记和预测。达到标签查询上限后, 对不纯的块, 采取投票的方式确定剩余未标记代表样本的标签。

主动学习阶段结束时, 代表样本均已获得标签。将代表样本作为训练集, 采用 kNN 算法对其他样本进行分类。

2.3　样例分析

提供一个样例分析来进一步清楚说明 PAL 算法。使用表 1 的决策信息系统, 允许查询的最大标签数 N=7。设置阻尼 γ=0.95, ε=0.01。图 2 和图 3 展示两次迭代聚类之后查询标签的情况。



(a) 二叉树

bl_1	9							
bl_2	11	13	14	15				
bl_3	6	7	8	10	12			
bl_4	0	1	2	3	4	5		

(b) 聚类信息块

图 2　第一次迭代

Fig. 2　First iteration of the running example

bl_1	9							
bl_2	12							
bl_3	11	13	14	15				
bl_4	6	7	8	10				
bl_5	0	1	2	3	4	5		

图 3　第二次迭代

Fig. 3　Second iteration of the running example

1) 根据算法 1 得到排名向量

Rank=[1 2 3 7 14 13 6 8 10 11 15 0 4 12 5 9]

根据算法 2 生成二叉树, 如图 2(a) 所示。由于数据集极小, 设定 R=100。令标记集合 U_1=Ø, 预测集合 U_2=Ø, 未分类集合 U_3=U。

2) 选择节点间的较低相似度 0.196 作为当前阈值 threshold, 根据算法 3 聚类可得块信息 bl = [bl_1 $bl_2 \cdots bl_4$]; 如图 2(b) 所示。接着, 查询 bl_1、bl_2、bl_3 和 bl_4 中样本 x_9、x_{11}、x_6 和 x_0 的标签, 分别是 iv、it、iv 和 is。在此次迭代后, $U_1' = \{x_0, x_6, x_9, x_{11}\}$, $U_1 = \varnothing \cup U_1' = \{x_0, x_6, x_9, x_{11}\}$, $U_3 = U_3 - U_1 = \{x_1, x_2, x_3, x_4, x_5, x_7, x_8, x_{10}, x_{12}, x_{13}, x_{14}, x_{15}\}$; 在第一次迭代后, $|U_1| \leqslant 7$。增大阈值 threshold进行第 2 次聚类。

3) 设置阈值 threshold = 0.323, 聚类簇信息块如图 3 所示。查询 bl_1、bl_2、bl_3 和 bl_4 中样本 x_{12}、x_{13} 和 x_7 的标签, 分别是 it、it 和 iv 。此时, $U_1' = \{x_{12}, x_{13}, x_7\}$, $U_1 = U_1 \cup U_1' = \{x_0, x_6, x_7, x_9, x_{11}, x_{12}, x_{13}\}$。$bl_2$ 中已被标记的样本数量已经足够多, 且已标记样本标签均为 it, 便可预测 bl_2 中剩余样本 x_{14} 和 x_{15} 的标签为 it。同理可预测块 bl_3 中剩余样本 x_8、x_{10} 的标签均为 iv。$U_2 = U_2 \cup U_2' = \{x_8, x_{10}, x_{14}, x_{15}\}$, $U_3 = U_3 - U_1 - U_2 = \{x_1, x_2, x_3, x_4, x_5\}$。在第 2 次迭代后, $|U_1| \geqslant 7$。不再进行第 3 次聚类。

4) $U_3 = \{x_1, x_2, x_3, x_4, x_5\}$ 通过投票给予标签;

bl_4 中 x_0 被标记为 is。所以 x_1、x_2、x_3、x_4 和 x_5 被标记为 is。

在本例中，查询 7 个样本的标签，预测 4 个样本的标签，5 个样本通过投票获得标签。无样本被错误标记，因此，精度为 100%。

3　实验及分析

在本节中，通过实验将 PAL 算法与传统的分类算法、主动学习算法进行比较，并回答以下问题：

1) PAL 算法选择代表样本是否具有可靠性，尤其不同二叉树比例 R 的设置对精度的影响；

2) PAL 算法是否比其他监督学习算法更精确；

3) PAL 算法是否比主动学习算法的分类效果好。

3.1　实验步骤

实验结合 Weka，在 macOS Sierra 操作系统下运行，其硬件配置为：2.6 GHz Intel Core i5 处理器，8 GB 1600 MHz DDR3。

实验采用 8 个公开的数据集，并将 PAL 算法与 J48、kNN、Naïve Bayes、One-R 和 Logistics[18] 这 5 种传统的监督学习算法进行比较，同时与 QBC、KQBC 和 MADE 这 3 种主动学习算法作对比。实验采用分类精度 accuracy 作为评估指标。

与传统的监督学习分类算法的比较实验中，针对每个数据集，实验设置训练集以 1% 为步长，规模由 1% 增加到 10%。在训练集规模不同的情况下，观察分类精度的变化。在与主动学习算法的比较实验中，训练集规模均设置为 10%。

设置二叉树比例 $R \in [20\%, 50\%]$，阻尼因子 $\gamma \in [0.65, 0.95]$，极小值 ε=0.01。为了降低实验的随机性误差，采用相同参数设置进行 10 次重复实验，取得平均值作为实验结果。

实验所用数据集详细信息如表 3 所示。

表 3　数据集描述

Table 3　Description of experimental datasets

数据集	条件属性	决策属性	数量
Iris	4	3	150
Flame	2	2	240
E.coli	7	8	336
Seeds	7	3	210
Diabetes	8	2	768
Jain	2	2	373
Aggregation	2	7	788
Twonorm	20	2	7 400

3.2　参数 R 对分类效果的影响

在本节中，将回答问题 1)。讨论不同的二叉树比例 R 对实验精度的影响。表 4 展现了在训练集规模是数据集的 10% 的情况下，所得精度随 R 的变化情况。

表 4　PAL 算法在不同二叉树构建比例 *R* 下分类精度的比较

Table 4　Classification accuracy comparisons of PAL based on different Binary Tree ratios *R*

数据集	20%	30%	40%	50%	60%	70%	80%	90%	100%
Iris	0.963 0	0.955 6	0.955 6	0.844 4	0.822 2	0.822 2	0.800 0	0.792 6	0.822 2
Flame	0	0.981 5	0.986 1	0.990 7	0.907 4	0.879 6	0.898 1	0.856 5	0.953 7
E.coli	0.749 2	0.808 6	0.821 8	0.808 6	0.765 7	0.742 6	0.736 0	0.709 6	0.745 9
Seeds	0.900 0	0.894 2	0.883 6	0.878 3	0.873 0	0.888 9	0.888 9	0.629 6	0.899 5
Diabetes	0.648 8	0.687 9	0.715 3	0.693 6	0.651 7	0.650 3	0.648 8	0.644 5	0.696 5
Jain	1.000 0	1.000 0	0.994 0	0.988 1	0.979 2	0.979 2	0.973 2	0.967 3	1.000 0
Aggregation	1.000 0	0.985 9	0.993 0	0.922 5	0.918 3	0.932 4	0.973 2	0.964 8	0.956 3
Twonorm	0.971 0	0.965 3	0.961 6	0.959 9	0.953 0	0.938 4	0.930 0	0.927 6	0.956 0

由表 4 可以看出，对于不同的数据集，最佳的二叉树比例取值存在差异。但从整体来看，最佳取值都集中在[20, 50]区间。

实验结果符合数据集样本的分布规律，信息量较高的样本所占的比例较小。二叉树比例取值越大时，越多的信息量低的样本参与到二叉树的构建，一些离群点、边界点影响聚类结果，而导致分类错误。同时表明，将 PageRank 分值作为样本信息量的度量指标具有可靠性。

Iris、Seeds、Twonorm 数据集样本均匀，不存在样本倾斜问题，二叉树聚类算法能够获得很好的分簇效果，因此二叉树比例取值较小时，能够保证查询到的样本都具有高信息量，反而分类精度更高。

较大数据集，如 Twonorm、Aggregation，R 比例较小，所选代表样本构成的树形结构也能很好地表现样本的层次结构，因此对分类精度不会有较大影响。

在本文后续的研究讨论中，R 当作经验参数参与二叉树的构建。

3.3　与经典算法对比

在本节中，将回答第二个问题。PAL 在 8 个

数据集上与 J48、Naïve Bayes、kNN、One-R 和 Logistics 经典算法做了对比。图 4 展示了 PAL 算法以及对比算法在不同训练集比例下的分类精度变化趋势。

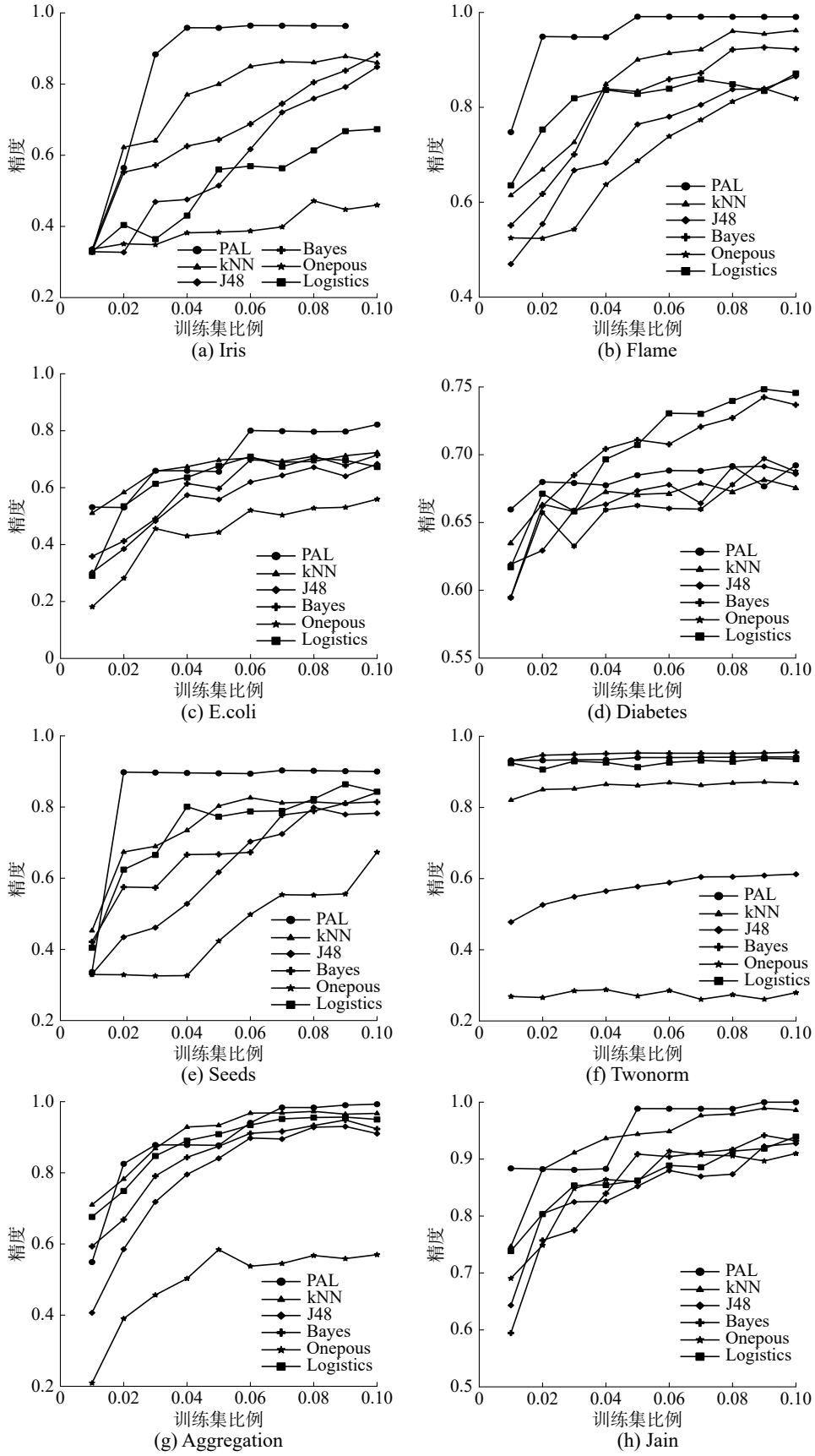


图 4　与经典算法对比

Fig. 4　Comparison with classical algorithms

实验结果表明，本文提出的 PAL 算法在 Iris、Flame、Ecoli、Seeds、Aggregation 和 Jain 数据集上，分类精度高于对比的经典算法，尤其是 Flame 数据集，在实验所选的训练集比例下，分类精度均高于经典分类算法。在 Twonorm 数据集上也能取得较好的分类精度，分类精度达到 97%，仅略低于 Naïve Bayes 算法。在 Diabetes 数据集上优势不明显，尤其是在 Diabetes 数据集上，PAL 算法分类精度高于 kNN、J48 和 One-R，但是低于 Naïve Bayes 和 Logistics。

图 4(b)、(d)、(f) 显示，在实验所选的所有训练集规模下，对应数据集上 PAL 算法分类精度均高于 kNN 算法；图 (a)、(c)、(e)、(g)、(h) 显示，在多数训练集规模下，对应数据集上 PAL 算法分类精度高于 kNN 算法。PAL 对代表样本采用主动学习算法进行标记和预测，而对于剩余样本则采用 kNN 进行预测。因此，当二叉树比例 R=0 时，PAL 算法将退化成 KNN 算法。该结果表明，PAL 的样本选择策略和主动学习算法具有可行性。

图 4(a)、(b)、(d)、(e)、(h) 显示，在训练集规模极小的情况下，如 R=10% 时，PAL 较其他经典算法能取得较好的分类精度；图 4(a)、(b)、(e)、(g) 显示，训练集规模为 30% 之前，PAL 算法的分类精度快速地上升，逐渐趋于稳定，说明 PageRank 分值作为样本信息量的度量指标具有可靠性，结合聚类算法，利用样本的分布信息能够有效地进行样本选择。

图 4(a)、(b)、(e)、(f) 显示，在 Iris、Flame、Seeds 数据集上分类时，训练集的规模对 PAL 分类精度影响不明显，是因为数据集太小，训练集比例对分类效果影响较低。在 Twonorm 数据集上，训练集的规模对所有算法的分类精度影响均不明显，说明在该数据集上数据分布较为均匀。

3.4　与主动学习算法对比

将 PAL 算法与流行的 3 种主动算法进行比较。表 5 展现了在训练集规模是数据集的 10%，R 设置为 40% 的情况下，QBC、KQBC、MADE 和 PAL 的分类精度。为了更清晰地展示各个算法的性能差异，设计以排名为衡量标准的评估方法。

从总体上看，本文提出的 PAL 算法与其他主动学习算法比较平均排名靠前。PAL 算法在 Iris、Flame、Seeds、Diabetes 和 Twonorm 数据集上，分类精度高于其他对比的主动学习算法，尤其在 Flame 数据集上，分类精度达到 98%。在 Ecoli、Jain 和 Aggregation 数据集上也有很好的分类表现。

表 5　PAL 与 3 种主动学习算法的比较

Table 5　Accuracies of PAL and three active learning algorithms

数据集	QBC	KQBC	MADE	PAL
Iris	0.937 0	0.899 8	0.918 5	0.955 6
Flame	0.946 3	0.714 4	0.986 1	0.986 1
Ecoli	0.829 7	0.591 0	0.715 2	0.821 8
Seeds	0.876 7	0.845 8	0.857 1	0.883 6
Diabetes	0.708 8	0.647 9	0.688 9	0.715 3
Jain	0.960 7	0.918 2	1.000 0	0.994 0
Aggregation	0.891 7	0.627 2	0.994 4	0.993 0
Twonorm	0.902 3	0.942 9	0.952 7	0.961 6
AveRank	2.500 0	3.900 0	2.100 0	1.400 0

4　结束语

本文提出了一种基于 PageRank 的主动学习算法，为样本的选择问题提供了一种可行的方案。利用 PageRank 理论发现信息量较高的代表样本，从而在该集群上构建二叉树，用来表示样本的层次结构。在二叉树上进行迭代聚类，标记和预测，能够保证查询到的样本分布均匀，同时避免离群点的影响。用代表对象训练得到分类模型，采用 kNN 算法处理剩余样本。实验结果表明，PAL 算法相比于 Naïve Bayes 和 J48 等传统分类算法，能得到更高的分类精度，且与 QBC 等主动学习算法相比，分类效果更好。

参考文献：

[1] MINN S, 傅顺开, 吕天依, 等. 一般贝叶斯网络分类器及其学习算法[J]. 计算机应用研究, 2016, 33(5): 1327–1334.
MINN S, FU Shunkai, LV Tianyi, et al. Algorithm for exact recovery of Bayesian network for classification[J]. Application research of computer, 2016, 33(5): 1327–1334.

[2] 王翔, 胡学钢, 杨秋洁. 基于 One-R 的改进随机森林入侵检测模型研究[J]. 合肥工业大学学报 (自然科学版), 2015, 38(5): 627–630, 711.
WANG Xiang, HU Xuegang, YANG Qiujie. Research on improved intrusion detection model with random forest based on feature evaluation of One-R[J]. Journal of Hefei University of Technology (natural science), 2015, 38(5): 627–630, 711.

[3] YANG Yi, CHEN Wenguang. Taiga: performance optimization of the C4.5 decision tree construction algorithm[J]. Tsinghua science and technology, 2016, 21(4): 415–425.

[4] ZHOU Xueyuan, BELKIN M. Semi-supervised learning[J]//Journal of the royal statistical society, 2010, 172(2): 530.

[5] WANG Min, MIN Fan, ZHANG Zhiheng, et al. Active learning through density clustering[J]. Expert systems with applications, 2017, 85: 305–317.

[6] 胡小娟, 刘磊, 邱宁佳. 基于主动学习和否定选择的垃圾邮件分类算法[J]. 电子学报, 2018, 46(1): 203–209.
HU Xiaojuan, LIU Lei, QIU Ningjia. A novel spam categorization algorithm based on active learning method and negative selection algorithm[J]. Acta electronica sinica, 2018, 46(1): 203–209.

[7] SYED A R, ROSENBERG A, KISLAL E. Supervised and unsupervised active learning for automatic speech recognition of low-resource languages[C]// Proceedings of 2016 IEEE International Conference on Acoustics, Speech and Signal Processing. Shanghai, China, 2016: 5320–5324.

[8] SUN Shujin, ZHONG Ping, XIAO H, et al. An MRF model-based active learning framework for the spectral-spatial classification of hyperspectral imagery[J]. IEEE journal of selected topics in signal processing, 2015, 9(6): 1074–1088.

[9] YANG Yi, MA Zhigang, NIE Feiping, et al. Multi-class active learning by uncertainty sampling with diversity maximization[J]. International journal of computer vision, 2015, 113(2): 113–127.

[10] XIONG Sicheng, AZIMI J, FERN X Z. Active learning of constraints for semi-supervised clustering[J]. IEEE transactions on knowledge and data engineering, 2014, 26(1): 43–54.

[11] BLOODGOOD M. Support vector machine active learning algorithms with query-by-committee versus closest-to-hyperplane selection[C]//Proceedings of 2018 IEEE 12th International Conference on Semantic Computing. Laguna Hills, USA, 2018: 148–155.

[12] BRIN SERGEY, PAGE Lawrence. The anatomy of a large-scale hypertextual web search engine [J]. Computer networks and ISDN systems, 1998, 30(1/7): 107-117.

[13] DENG Zhenyun, ZHU Xiaoshu, CHENG Debo, et al. Efficient kNN classification algorithm for big data[J]. Neurocomputing, 2016, 195: 143–148.

[14] GILAD-BACHRACH R, NAVOT A, TISHBY N. Kernel query by committee (KQBC)[R]. Technical Report 2003–88, Leibniz Center, the Hebrew University, 2003.

[15] CAI Deng, HE Xiaofei. Manifold adaptive experimental design for text categorization[J]. IEEE transactions on knowledge and data engineering, 2012, 24(4): 707–719.

[16] MIN Fan, ZHU W. A competition strategy to cost-sensitive decision trees[C]//Proceedings of the 7th International Conference on Rough Sets and Knowledge Technology. Chengdu, China, 2012: 359–368.

[17] 张桃, 吴小伟. 基于 PageRank 的马尔可夫链研究[J]. 电子设计工程, 2017, 25(9): 36–38.
ZHANG Tao, WU Xiaowei. The study of Markov chains based on PageRank[J]. Electronic design engineering, 2017, 25(9): 36–38.

[18] LIU Dun, LI Tianrui, LIANG Decui. Incorporating logistic regression to decision-theoretic rough sets for classifications[J]. International journal of approximate reasoning, 2014, 55(1): 197–210.

作者简介:

邓思宇, 女, 1993 年生, 硕士研究生, 主要研究方向为代价敏感学习、主动学习。

刘福伦, 男, 1993 年生, 硕士研究生, 主要研究方向为代价敏感学习、粗糙集、主动学习。

黄雨婷, 女, 1996 年生, 主要研究方向为推荐系统。