Journal of System Simulation

Volume 36 | Issue 5 Article 13

5-15-2024

Tri-training Algorithm Based on Density Peaks Clustering

Yuhang Luo
School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, China, 1658051291@qq.com

Runxiu Wu
School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, China

Zhihua Cui
College of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China

Yiying Zhang
College of Artificial Intelligence, Tianjin University of Science & Technology, Tianjin 300457, China

See next page for additional authors

Tri-training Algorithm Based on Density Peaks Clustering

Abstract

Abstract: Tri-training can effectively improve the generalization ability of classifiers by using unlabeled data for classification, but it is prone to mislabeling unlabeled data, thus forming training noise. Tritraining (Tri-training with density peaks clustering, DPC-TT) algorithm based on density peaks clustering is proposed. The DPC-TT algorithm uses the density peaks clustering algorithm to obtain the class cluster centers and local densities of the training data, and the samples within the truncation distance of the class cluster centers are identified as the samples with better spatial structure, and these samples are labeled as the core data, and the classifier is updated with the core data, which can reduce the training noise during the iteration to improve the performance of the classifier. The experimental results show that the DPC-TT algorithm has better classification performance compared with the standard Tri-training algorithm and its improvement algorithm.

Keywords

Tri-training, semi-supervised learning, density peaks clustering, spatial structure, classifier

Authors

Yuhang Luo, Runxiu Wu, Zhihua Cui, Yiying Zhang, Yeshen He, and Jia Zhao

Recommended Citation

Luo Yuhang, Wu Runxiu, Cui Zhihua, et al. Tri-training Algorithm Based on Density Peaks Clustering [J]. Journal of System Simulation, 2024, 36(5): 1189-1198.

第 36 卷第 5 期 系统仿真学报© Vol. 36 No. 5
2024 年 5 月 Journal of System Simulation May 2024

基于密度峰值聚类的Tri-training算法

罗宇航[1]，吴润秀[1]，崔志华[2]，张翼英[3]，何业慎[4]，赵嘉[1*]
(1. 南昌工程学院 信息工程学院，江西 南昌 330099；2. 太原科技大学 计算机科学与技术学院，山西 太原 030024；
3. 天津科技大学 人工智能学院，天津 300457；4. 深圳市国电科技通信有限公司，广东 深圳 518000)

摘要：Tri-training利用无标签数据进行分类可有效提高分类器的泛化能力，但其易将无标签数据误标，从而形成训练噪声。提出一种基于密度峰值聚类的Tri-training(Tri-training with density peaks clustering，DPC-TT)算法。*密度峰值聚类通过类簇中心和局部密度可选出数据空间结构表现较好的样本。DPC-TT算法采用密度峰值聚类算法获取训练数据的类簇中心和样本的局部密度，对类簇中心的截断距离范围内的样本认定为空间结构表现较好，标记为核心数据，使用核心数据更新分类器，可降低迭代过程中的训练噪声，进而提高分类器的性能。*实验结果表明：相比于标准Tri-training算法及其改进算法，DPC-TT算法具有更好的分类性能。

关键词：Tri-training；半监督学习；密度峰值聚类；空间结构；分类器
中图分类号：TP391.9；TP18　　文献标志码：A　　文章编号：1004-731X(2024)05-1189-10
DOI: 10.16182/j.issn1004731x.joss.22-1550
引用格式: 罗宇航, 吴润秀, 崔志华, 等. 基于密度峰值聚类的Tri-training算法[J]. 系统仿真学报, 2024, 36(5): 1189-1198.
Reference format: Luo Yuhang, Wu Runxiu, Cui Zhihua, et al. Tri-training Algorithm Based on Density Peaks Clustering [J]. Journal of System Simulation, 2024, 36(5): 1189-1198.

Tri-training Algorithm Based on Density Peaks Clustering

Luo Yuhang[1], *Wu Runxiu*[1], *Cui Zhihua*[2], *Zhang Yiying*[3], *He Yeshen*[4], *Zhao Jia*[1*]
(1. School of Information Engineering, Nanchang Institute of Technology, Nanchang 330099, China; 2. College of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China; 3. College of Artificial Intelligence, Tianjin University of Science & Technology, Tianjin 300457, China; 4. China Gridcom Co., Ltd., Shenzhen 518000, China)

Abstract: Tri-training can effectively improve the generalization ability of classifiers by using unlabeled data for classification, but it is prone to mislabeling unlabeled data, thus forming training noise. Tri-training (Tri-training with density peaks clustering, DPC-TT) algorithm based on density peaks clustering is proposed. *The DPC-TT algorithm uses the density peaks clustering algorithm to obtain the class cluster centers and local densities of the training data, and the samples within the truncation distance of the class cluster centers are identified as the samples with better spatial structure, and these samples are labeled as the core data, and the classifier is updated with the core data, which can reduce the training noise during the iteration to improve the performance of the classifier.* The experimental results show that the DPC-TT algorithm has better classification performance compared with the standard Tri-training algorithm and its improvement algorithm.

Keywords: Tri-training; semi-supervised learning; density peaks clustering; spatial structure; classifier

收稿日期：2022-12-29　　修回日期：2023-03-11
基金项目：国家自然科学基金(52069014)
第一作者：罗宇航(1997-)，男，硕士生，研究方向为数据挖掘。E-mail：1658051291@qq.com
通讯作者：赵嘉(1981-)，男，教授，博士，研究方向为复杂系统建模与优化、智能计算与计算智能、大数据与深度学习等。
E-mail：zhaojia925@163.com

0 引言

数据挖掘指从大量的数据中搜索隐藏于其中的模式和规律，为后续的预测和决策提供依据，挖掘方法包括监督学习、无监督学习和半监督学习3类。监督学习用大量标签数据训练分类器来保证模型良好的性能。无监督学习无需先验信息，采用无标签数据将相似性样本聚集在一起生成聚类模型，模型的准确性难以保证。大数据时代，数据量增长迅速，获取无标签数据较为容易，相比之下，有标签数据的获取需耗费大量人力、物力和财力。半监督学习结合监督学习和无监督学习的优势，运用大量无标签数据与少量有标签数据进行学习，目前已广泛应用于图像处理[1]、医学诊断[2]、虚假评论检测[3]、网络安全[4]、异常检测[5]等领域。

半监督学习[6-9]从不同的学习场景出发可分为4类：半监督分类、半监督回归、半监督聚类以及半监督降维。其中，半监督分类学习方法使用无标签数据训练分类模型，在消耗较小的资源下可获得不弱于监督分类模型的性能。主要的半监督分类方法有基于分歧的方法[10]、生成式方法[11]、基于图的方法[12]和判别式方法[13]等。

基于分歧的分类算法起源于Co-training算法[10]。Co-training算法[14]通过2个分类器相互训练提高分类器性能，算法要求有标签数据具有2个充分冗余且条件独立的视图，但大多数情况下，训练数据通常不满足充分冗余的条件。为解决数据集的充分冗余问题，文献[15]在协同训练算法的基础上提出了Tri-training算法。该算法不要求训练数据满足充分冗余的视图，有效降低了分类算法对数据的要求，提高了分类算法的效率和泛化能力。

Tri-training算法使用3个基分类器对无标签数据添加伪标签进行训练，同大多数半监督分类算法一样，Tri-training算法容易对无标签数据进行误标。如何选择更具代表性的无标签数据来提高分类模型的性能是Tri-training算法需要解决的问题，同样也是所有半监督分类算法需要解决的重要问题。针对上述问题，文献[16]引入自适应数据剪辑策略，识别并移除每次迭代过程中的误标签样本，通过减少误标样本的数量提升分类器性能。文献[17]将集成学习与Tri-training算法结合，通过集成学习对无标签数据的置信度进行估计来添加伪标签，提高了分类模型的性能，同时算法时间复杂度低。文献[18]将交叉熵和凸优化思想引入Tri-training算法，运用交叉熵代替分类错误率反映预测分布与真实分布之间的差异，并使用凸优化对分类器赋予权重降低无标签数据的预测误差，达到降低标签噪声的效果。文献[19]提出自适应密度剪辑与交叉熵评估的Tri-training算法，通过最近邻剪辑结合样本局部密度剔除训练噪声，优化模型性能。文献[20]提出基于DECORATE集成学习与置信度评估的Tri-training算法，通过集成学习添加差异化人工样本训练基分类器，同时对每一轮标记样本进行置信度评估，有效减少噪声数据的影响。文献[21]提出一种结合聚类和分类的半监督学习框架，通过模糊C均值聚类选取信息量高的无标签样本辅助训练分类器。文献[22]使用密度峰值聚类算法找到无标签数据的空间结构，将数据的空间结构融入到自训练迭代过程中，有效减少了训练噪声。综上所述，学者们对Tri-training算法进行改进时大多只采用减少训练过程中训练噪声的方式来提高分类器性能，若迭代初期通过Bootstrap抽样大多为信息量低的数据时，分类器效果并不理想。文献[21-22]表明，聚类方法可以辅助找寻信息量高的无标签数据，达到减少迭代过程中样本易被误标的目的。

密度峰值聚类(density peaks clustering, DPC)[23]是一种基于密度的聚类算法，可自动寻优找到类簇中心，能探索任何簇结构数据集的全局空间结构，已成为聚类分析的研究热点之一[24]。为了更好地选择隐含信息量高的无标签数据，减少迭代过程中样本误标数量，将DPC与Tri-training算法结合，通过对数据集进行聚类，找出能较好表现

DOI: 10.16182/j.issn1004731x.joss.22-1550

数据空间结构的样本标记为核心数据进行噪声过滤，分类器采用核心数据进行训练学习，提出基于密度峰值聚类的Tri-training(DPC-TT)算法。

1 相关知识

1.1 Tri-training算法

Tri-training算法不同于Co-training算法，对数据集的视图不做要求，算法使用少量有标签数据集训练3个基分类器，用训练好的基分类器联合投票对无标签数据进行预测并添加伪标签，将添加伪标签的数据和初始带标签数据结合重新训练基分类器，最后分类器联合投票来预测所求样本的类标签。首先，假设初始有少量带标签的数据集L和大量无标签数据集U，通过Bootstrap对L进行随机抽样，训练3个不同的基分类器h_1、h_2和h_3。之后，基分类器使用投票法对数据集U中的数据x添加伪标签，如基分类器h_1、h_2对x的预测结果一致，则将x加入到基分类器h_3的训练集中。基分类器h_1，h_2的训练集按照相同原理进行扩充，使用扩充后的训练集对基分类器h_1、h_2和h_3重新训练。根据可学习理论可知，随着训练集增加，分类器的错误率会降低，反复迭代直至分类器h_1、h_2和h_3的错误率不再发生变化，训练过程结束。最后，采用多数投票法确定最终的分类结果。

1.1.1 可学习理论

根据文献[25]的结论，当抽取样例数量为m的序列σ，样例数量满足式(1)时，最小化H_i与σ不一致，具有PAC可学习性，即$\Pr[d\left(H_i,H^*\right)\geqslant\epsilon]\leqslant\delta$。

$$m\geqslant\frac{2}{\epsilon^2\left(1-2\eta\right)^2}\ln\left(2N/\delta\right) \tag{1}$$

式中：ϵ为分类错误率的上限；η为训练集噪声率的上限$\left(\eta<0.5\right)$；N为假设数；δ为置信度。

假设$m'=\frac{2}{\epsilon^2\left(1-2\eta\right)^2}\ln\left(2N/\delta\right)$，且$m/m'=\mu$，通过常量$c=2\mu\ln\left(2N/\delta\right)$，式(1)可变为

$$m=\frac{c}{\epsilon^2\left(1-2\eta\right)^2} \tag{2}$$

所以，分类错误率ϵ与训练集噪声率η、训练集规模m有如下关系：

$$c/\epsilon^2=m\left(1-2\eta\right)^2 \tag{3}$$

即，$m\left(1-2\eta\right)^2\propto\frac{1}{\epsilon^2}$

Tri-training算法可学习理论：基于式(3)，保证训练集规模m增加的同时，确保重新训练的分类器错误率会逐渐降低。

1.1.2 Tri-training训练集更新条件

Tri-training每经过一轮迭代，需要对分类器的训练集进行扩充，任意两个基分类器h_1、h_2会从$\boldsymbol{U}$中挑选出一部分数据来添加伪标签，添加完伪标签的数据集在满足添加条件后加入第3个基分类器h_3的训练集中。

以扩充基分类器h_1的训练集为例，假设$\boldsymbol{L}^t$和$\boldsymbol{L}^{t-1}$分别代表第t轮和第t–1轮迭代后为h_1添加的训练集，在进行第t轮迭代时，t–1轮中添加的数据集$\boldsymbol{L}^{t-1}$会重新放回$\boldsymbol{U}$当中，最后第t轮和第t–1轮中的训练集分别为$\boldsymbol{L}\cup\boldsymbol{L}^t$和$\boldsymbol{L}\cup\boldsymbol{L}^{t-1}$，第$t$轮和第$t$–1轮中训练集样本的数量$m^t$和$m^{t-1}$分别为$|\boldsymbol{L}\cup\boldsymbol{L}^t|$和$|\boldsymbol{L}\cup\boldsymbol{L}^{t-1}|$。第$t$轮迭代中基分类器$h_1$的训练集噪声率为

$$\eta^t=\frac{\eta_L|\boldsymbol{L}|+e_1^t|\boldsymbol{L}^t|}{|\boldsymbol{L}\cup\boldsymbol{L}^t|} \tag{4}$$

式中：η_L为初始标签数据L上的噪声率；e_1^t为第t轮迭代中基分类器h_2和h_3组成的联合分类器错误率上限。

由可学习性理论可知，训练集规模的扩大可以降低分类错误率，在满足式(5)的约束条件下，分类器的性能会得到提升。

$$\begin{aligned}&|\boldsymbol{L}\cup\boldsymbol{L}^t|\left(1-2\frac{\eta_L|\boldsymbol{L}|+e_1^t|\boldsymbol{L}^t|}{|\boldsymbol{L}\cup\boldsymbol{L}^t|}\right)^2>\\&|\boldsymbol{L}\cup\boldsymbol{L}^{t-1}|\left(1-2\frac{\eta_L|\boldsymbol{L}|+e_1^{t-1}|\boldsymbol{L}^{t-1}|}{|\boldsymbol{L}\cup\boldsymbol{L}^{t-1}|}\right)^2\end{aligned} \tag{5}$$

由式(5)可知，当$|\boldsymbol{L}^t|>|\boldsymbol{L}^{t-1}|$时，可得$e_1^t|\boldsymbol{L}^t|>$

$e_1^{t-1}|\boldsymbol{L}^{t-1}|$，一般假定错误率上限为$0 \leqslant e_1^t \leqslant 0.5$，即有

$$0 < e_1^t / e_1^{t-1} < |\boldsymbol{L}^{t-1}| / |\boldsymbol{L}^t| < 1 \tag{6}$$

在 Tri-training 算法迭代过程中，若数据集规模满足式(6)，则h_2、h_3标签的伪标签样本$\boldsymbol{L}^t$将加入到h_1的新训练集中，反之，则h_1的训练集不发生变化。分类器h_2、h_3训练集运用同理进行扩充。

1.2 密度峰值聚类算法

密度峰值聚类算法是由 Alex 等[23]提出的一种聚类方法，DPC 算法需满足以下条件：①类簇中心的密度较大；②任意两个类簇中心的距离较远。算法通过局部密度和相对距离组成的决策图确定类簇中心，之后对剩余样本进行分配。

DPC 算法中有 2 种方式定义局部密度：截断核和高斯核。当数据集规模较大时，局部密度通常使用截断核的方式定义

$$\rho_i = \sum_{i \neq j} \chi\left(d_{ij} - d_{\mathrm{c}}\right), \chi(x) = \begin{cases} 1, x < 0 \\ 0, x \geqslant 0 \end{cases} \tag{7}$$

当数据集规模较小时，局部密度使用高斯核的方式进行定义：

$$\rho_i = \sum_{i \neq j} \exp\left[-\frac{d_{ij}}{d_{\mathrm{c}}}\right]^2 \tag{8}$$

式中：d_{ij}为样本i和样本j之间的欧式距离；d_{c}为样本的邻域截断距离。

根据文献[23]可知，样本i的相对距离定义为

$$\begin{cases} \delta_i = \min\limits_{j:\rho_j > \rho_i} \left(d_{ij}\right) \\ \delta_i = \max\limits_{i \neq j} \left(\delta_j\right) \end{cases} \tag{9}$$

通过求解各数据的局部密度和相对距离构建决策图，选择局部密度高且相对距离大的样本点作为类簇中心，将剩余点分配给密度比它高且距离它最近样本的所属类簇，完成聚类过程。

2 基于密度峰值聚类的 Tri-training 算法

2.1 算法思想

Tri-training 算法添加伪标签的过程中，容易对无标签数据进行误标。数据在采集处理过程中可能会有一些错误信息，导致一些样本与其他样本间的数据信息差异过大。如果将这些差异较大的样本适当地进行过滤，Tri-training 在添加伪标签时便可降低误标率，但数据样本不能大部分进行过滤，若能找到一种方法能将正确的数据进行保留，错误的数据进行去除，则能提高分类器性能。根据文献[22]的结论，DPC 算法能找到整个数据空间的真实结构，故可设定一定阈值，通过 DPC 算法找出能较好表现数据空间结构的样本并定为核心数据，其余为非核心数据。非核心数据由于聚类效果不好，对其添加伪标签时易产生错误，同时，拥有更多错误信息的数据更大可能在非核心数据集中。因此，可通过 DPC 算法寻找数据集的核心数据，之后使用核心数据对 Tri-training 算法进行训练来提高分类器性能。基于以上分析，将密度峰值聚类算法与 Tri-training 算法结合，提出基于密度峰值聚类的 Tri-training 算法。将类簇中心的截断距离范围内的样本定义为核心数据：

$$\boldsymbol{I} = \left\{i | d_{ic} \leqslant d_{\mathrm{c}}, i \in (1, 2, \cdots, u), c \in (1, 2, \cdots, k)\right\} \tag{10}$$

$$\boldsymbol{U}' = \left\{x | x \subseteq x_I\right\} \tag{11}$$

式中：d_{ic}为类簇中心c与其他样本i之间的距离；d_{c}参数设置为距离矩阵中最小的 5% 点中的最大值作为截断距离d_{c}的阈值。扩充核心数据的具体步骤如下。

步骤 1：无标签数据归一化，计算样本间欧式距离，设置截断距离d_{c}；

步骤 2：计算样本局部密度ρ_i和相对距离δ_i；

步骤 3：挑选局部密度高且相对距离大的密度峰值作为簇中心；

步骤 4：将簇中心的截断距离范围内的样本定义为核心数据U'。

图 1 是一个含有 200 个样本的二维分布图，以不同的颜色区分其类别，黑色五角星代表的是通过 DPC 算法聚类后的类簇中心。从图 1 可以看出，靠近类簇中心时样本显得更密集，越往外越稀疏，

DOI: 10.16182/j.issn1004731x.joss.22-1550

样本间的密集程度也反映出样本的相似程度。保留相似程度高的样本，在不破坏数据大体结构的同时可以去除空间结构表现差的样本。图 1 中，圈内样本为算法处理后保留的核心数据，其余为非核心数据，非核心数据和核心数据间的样本相似性过大，容易导致分类器添加伪标签时进行误标，因此，将非核心数据当成噪声过滤可以减少误标的样本数量，提高分类模型的性能。

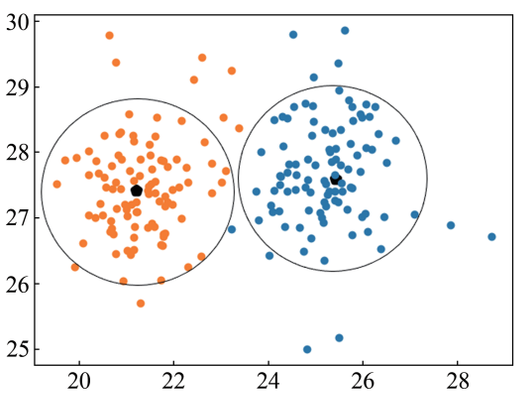


图 1　噪声过滤示意图
Fig. 1　Noise Filter Diagram

DPC-TT 算法的基本过程：首先，使用 DPC 算法对无标签数据集 $\boldsymbol{U}$ 进行聚类，求出各数据的局部密度和相对距离，得到数据集的类簇中心和样本的截断距离。之后，对数据集进行噪声过滤得到核心数据。最后，使用 Tri-training 算法对核心数据添加伪标签，训练分类器。

2.2 算法流程

DPC-TT 算法伪代码如下。

输入：无标签数据集 $\boldsymbol{U}$，有标签数据集 $\boldsymbol{L}$，测试集 $\boldsymbol{T}$

输出：测试集 $\boldsymbol{T}$ 中数据 x 的分类结果 $h(x)$

$c \leftarrow \varnothing; d_c \leftarrow 0; d_{ij} \leftarrow \varnothing$
$c, d_c, d_{ij} \leftarrow \mathrm{DPC}(U)$
for $i = 1 \sim u$ do
　if $(d_{ic} < d_c)$
　　$U' \leftarrow U_i$
end of for
for $i = 1 \sim 3$ do
　$S_i \leftarrow \mathrm{Bootstrap}(L), h_i \leftarrow \mathrm{Learn}(S_i)$
　$e_i' \leftarrow 0.5, l_i' \leftarrow 0$
end of for
repeat until none of $h_i (i \in \{1,2,3\})$ changes
　for $i = 1 \sim 3$ do
　　$L_i \leftarrow \varnothing$;update$_i \leftarrow$ FALSE
　　$e_i \leftarrow \mathrm{MeasureError}(h_j \& h_k)(j,k \neq i)$
　　if $(e_i < e_i')$
　　　then for every do
　　　if $h_i(x) = h_k(x)(j,k \neq i)$
　　　then $L_i \leftarrow L_i \cup \{(x, h_j(x))\}$
　　end of for
　　if $(l_i' = 0)$
　　　then $l_i' \leftarrow \left[\dfrac{e_i}{e_i' - e_i} + 1\right]$
　　if $(l_i' < |L_i|)$
　　then if $(e_i |L_i| < e_i' l_i')$
　　　then update$_i \leftarrow$ TRUE
　　　else if $l_i' > \dfrac{e_i}{e_i' - e_i}$
　　then $L_i \leftarrow \mathrm{Subsample}\left(L_i, \left[\dfrac{e_i' l_i'}{e_i} - 1\right] - 1\right)$
　　　update$_i \leftarrow$ TRUE
end of for
for $i = 1 \sim 3$ do
　if update$_i \leftarrow$ TRUE
　then $h_i \leftarrow \mathrm{Learn}(L \cup L_i); e_i' \leftarrow e_i$
$l_i' \leftarrow |L_i|$
end of for
end of repeat
Output:$h(x) \leftarrow \underset{y \in \text{label}}{\arg\max} \sum_{i:h_i(x)=y} 1$

上述伪代码中，第 1~6 步调用密度峰值聚类算法，对核心点进行标记并进行噪声过滤；第 7~10 步算法进行初始化操作，分别抽取 3 份初始训练集用于训练 3 个基分类器，分类器的初始错误

率设为0.5；第11~33步算法对噪声过滤后的无标签数据集添加伪标签来扩充训练集，用扩充后的训练集训练基分类器，直到分类器的错误率不再发生变化则训练过程结束；第34步算法通过3个基分类器联合投票来预测分类结果。

2.3 时间复杂度分析

Tri-training算法的时间复杂度主要由以下几部分构成：训练基分类器的时间复杂度$O(n)$；预测无标签数据的时间复杂度$O(n)$；训练集扩充的时间复杂度$O(n^2)$。综上，Tri-training算法的时间复杂度$O(n^2)$。DPC-TT算法的时间复杂度主要由以下部分构成：计算样本间欧氏距离的时间复杂度$O(n^2)$；计算样本局部密度的时间复杂度$O(n)$；核心数据扩充的时间复杂度$O(n)$；训练基分类器的时间复杂度$O(n)$；预测无标签数据的时间复杂度$O(n)$；训练集扩充的时间复杂度$O(n^2)$。综上，本文算法时间复杂度为$O(n^2)$，与Tri-training算法的时间复杂度处于同一量级。

3 实验结果与分析

3.1 实验设置

本文选用UCI机器学习库[26]中的数据集测试DPC-TT算法的性能。为验证该算法的有效性，选用Tri-training算法[15]、基于交叉熵的Tri-Training算法(TCE)[18]、安全的Tri-Training算法(ST)[18]，及基于交叉熵的安全Tri-Training算法(STCE)[18]进行比较。实验环境为Intel(R) Core(TM) i5-6300HQ CPU @ 2.30 GHz、12 G RAM、Windows10 64位操作系统和Python3.7编程环境。本文选取了9组UCI数据集进行实验，数据集的基本信息如表1所示。

3.2 实验结果分析

为模拟实际的数据采集过程和满足半监督分类学习的要求，本文实验选取UCI数据集中的80%做训练集，20%做测试集验证算法性能，在训练集中挑选20%为已标签数据集$\boldsymbol{L}$，80%为无标签数据集$\boldsymbol{U}$。

混淆矩阵可直观看出分类器识别的不同类元组。对此，本文在混淆矩阵的基础上对算法性能进行评价。如表2所示。

表1 实验数据集
Table 1 Experimental datasets

数据集	样本数	属性数	正类/%	负类/%
australian	690	14	44.5	55.5
wdbc	569	30	37.3	62.7
abalone	4 177	8	32.1	67.9
bupa	345	6	42.0	58.0
electrical	10 000	13	36.2	63.8
german	1 000	24	30.0	70.0
haberman	306	3	26.5	73.5
heart	270	13	44.4	55.6
spectf	267	44	20.6	79.4

表2 混淆矩阵
Table 2 Confusion matrix

真实类别	预测类别	
	正类	负类
正类	N_{TP}	N_{FN}
负类	N_{FP}	N_{TN}

表2中N_{TP}、N_{FP}、N_{FN}、N_{TN}分别表示真正例、假正例、假负例和真负例的样例数。本文用准确率、召回率、精度和F1值评价算法性能。准确率A反映模型对各类元组正确识别情况；召回率R反映分类模型查全能力；精度P为预测正类中实际为正元组的比例；F1值为召回率和精度的调和均值。

$$A=\left(N_{TP}+N_{TN}\right)/\left(N_{TP}+N_{FN}+N_{FP}+N_{TN}\right) \tag{12}$$

$$R=N_{TP}/\left(N_{TP}+N_{FN}\right) \tag{13}$$

$$P=N_{TP}/\left(N_{TP}+N_{FP}\right) \tag{14}$$

$$F_1=(2\times P\times R)/(P+R) \tag{15}$$

通过对9组UCI数据集进行实验，实验结果如表3~6所示。

图2是5种算法在9组数据集上的性能指标图。从表3~6和图2可以看出，在准确率、召回率、F1值3个评价指标上，DPC-TT算法的分类性能好。准确率和F1值指标上，DPC-TT有8组数据集上的效果表现最优；召回率上，有6组数据集上的效果表现最优。精度指标上，DPC-TT与STCE算法的分类效果差异不大，都在4组数据集上的表现最优。

表 3　准确率
Table 3　Accuracy

数据集	Tri-training	TCE	ST	STCE	DPC-TT
australian	0.802 2	0.820 6	0.826 6	0.849 7	**0.865 7**
wdbc	0.937 9	0.951 0	0.944 1	0.958 0	**0.972 1**
abalone	0.808 3	0.779 9	0.783 7	0.803 8	**0.823 0**
bupa	0.597 1	0.563 2	0.551 7	0.597 7	**0.645 6**
electrical	0.975 3	0.994 4	0.994 0	0.996 0	**0.999 5**
german	0.711 6	0.732 0	0.740 0	**0.746 0**	0.730 7
haberman	0.704 9	0.551 3	0.692 3	0.538 5	**0.754 1**
heart	0.763 2	0.720 6	0.764 7	0.779 4	**0.803 8**
spectf	0.641 5	0.641 8	0.567 2	0.626 9	**0.679 2**

表 4　召回率
Table 4　Recall

数据集	Tri-training	TCE	ST	STCE	DPC-TT
australian	0.742 2	0.776 5	0.807 7	0.782 6	**0.852 6**
wdbc	0.935 9	0.985 7	0.971 8	**1.000 0**	0.961 4
abalone	0.612 0	0.545 5	0.555 1	0.614 3	**0.752 2**
bupa	0.612 6	**0.800 0**	0.681 8	0.687 5	0.620 9
electrical	0.974 6	0.994 5	0.992 3	0.993 5	**0.999 0**
german	0.675 0	0.506 7	0.518 5	0.565 2	**0.747 1**
haberman	0.369 3	0.172 4	0.277 8	0.187 5	**0.454 2**
heart	0.716 5	0.647 1	0.709 7	0.705 9	**0.782 8**
spectf	**0.614 7**	0.342 9	0.277 8	0.342 1	0.586 2

表 5　F1 值
Table 5　F-measure

数据集	Tri-training	TCE	ST	STCE	DPC-TT
australian	0.749 0	0.809 8	0.807 7	**0.847 1**	0.777 5
wdbc	0.935 9	0.951 7	0.945 2	0.958 3	**0.961 4**
abalone	0.564 4	0.575 6	0.572 0	0.572 0	**0.768 3**
bupa	0.512 6	0.387 1	0.434 8	0.557 0	**0.620 8**
electrical	0.974 6	0.992 3	0.991 8	0.994 5	**0.999 1**
german	0.545 0	0.531 5	0.563 8	0.569 3	**0.747 0**
haberman	0.284 3	0.222 2	0.294 1	0.250 0	**0.378 8**
heart	0.741 0	0.698 4	0.733 3	0.761 9	**0.842 1**
spectf	0.545 8	0.500 0	0.408 2	0.509 8	**0.693 8**

表 6　精度
Table 6　Precision

数据集	Tri-training	TCE	ST	STCE	DPC-TT
australian	0.668 5	0.846 2	0.807 7	**0.923 1**	0.695 0
wdbc	0.886 6	0.920 0	0.920 0	0.920 0	**0.961 2**
abalone	**0.640 8**	0.609 4	0.589 8	0.535 2	0.636 8
bupa	0.327 6	0.255 3	0.319 1	**0.468 1**	0.316 5
electrical	0.957 7	0.990 2	0.991 3	0.995 6	**0.999 5**
german	0.883 3	0.558 8	0.617 6	0.573 5	**0.936 3**
haberman	0.233 3	0.312 5	0.312 5	**0.375 0**	0.329 4
heart	0.911 2	0.758 6	0.758 6	0.827 6	**0.911 4**
spectf	0.692 8	0.923 1	0.769 2	**1.000 0**	0.850 0

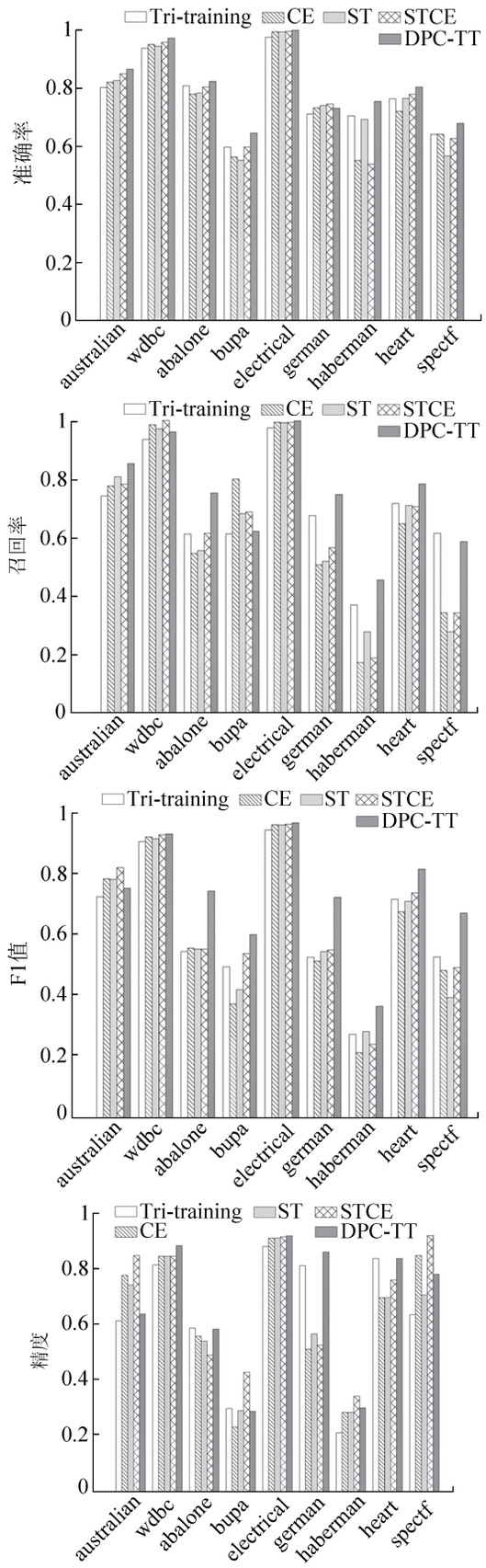


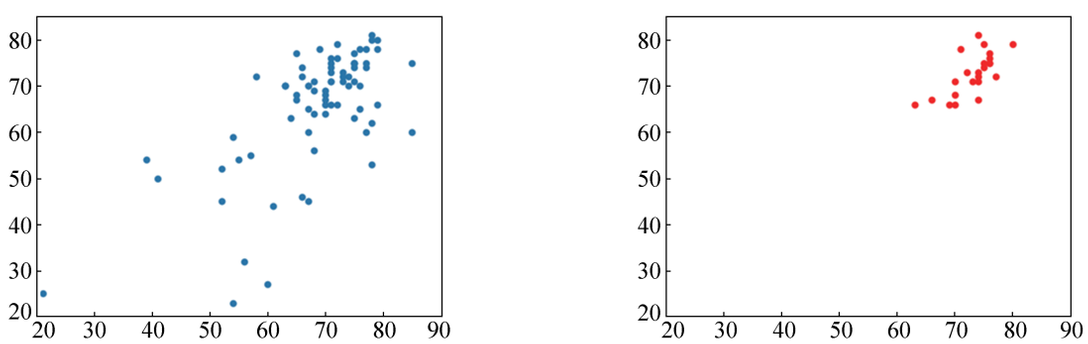
图 2　5 种算法在 9 组数据集上的性能指标
Fig. 2　Performance metrics of five algorithms on nine dataset

图2中，DPC-TT在部分数据集上的精度指标性能表现较差，核心数据筛选时聚类效果可能不好，导致分类的精度不高。为此，模拟数据集扩充机制，随机选取部分数据进行聚类实验，使用聚类评价指标FMI[27]进行评价，数值越接近1聚类效果越好，结果如表7所示。

从表7的实验结果可以得出聚类效果稍差的数据集精度指标表现不好。为进一步说明原因，随机从spectf数据集抽取100个样例正负元组分布图，如图3所示。从图3可以看出正负元组分布不平衡，同时密集程度分布也不均匀，正类元组相比负类元组更为稀疏，采用密度峰值聚类算法进行核心数据筛选时易使正类元组被剔除，导致聚类效果不好。

表7 FMI性能指标
Table 7 FMI Performance Indicators

数据集	FMI
australian	0.544 5
wdbc	0.745 5
abalone	0.560 6
bupa	0.511 2
electrical	0.533 0
german	0.761 3
haberman	0.567 7
heart	0.710 1
spectf	0.454 2



(a) 正类元组 (b) 负类元组

图3 数据集spectf正负元组二维图
Fig. 3 Dataset spectf positive and negative tuple bipartite graph

为评价各算法的综合性能，引入Friedman检验[28]对算法进行综合评价。Friedman检验的秩均值越大，表示该算法的性能更优。表8是各算法经Friedman检验的秩均值。

表8 Friedman检验后的平均等级
Table 8 Average rank after friedman test

评价指标	Tri-training	TCE	ST	STCE	DPC-TT
准确率	2.22	2.33	2.33	3.44	**4.67**
召回率	2.67	2.44	2.56	3.11	**4.22**
F1值	2.11	2.22	2.39	3.61	**4.67**
精度	2.44	2.44	2.56	3.39	**4.67**
均值	2.36	2.36	2.46	3.39	**4.56**

从表8可以看出，通过密度峰值聚类算法对数据集进行噪声过滤能提高分类器的性能。

4 结论

针对Tri-training算法在迭代过程中给无标签数据添加伪标签时容易误标的问题，本文提出了一种基于密度峰值聚类的Tri-training算法。该算法通过DPC算法对训练集进行聚类，将类簇中心的邻域截断距离范围内的样本标签为核心数据，对核心数据进行噪声过滤，进而降低训练集的噪声，减少了Tri-training算法迭代过程中无标签数据被误标的数量。实验结果表明：DPC-TT算法的分类性能有明显提升。然而，本文并未考虑数据的分布特点，下一步将结合数据的分布特点进行核心数据的挑选，提高核心数据选取的正确性。之后，可能会采用群体智能优化算法[29-30]来解决研究中的问题。

DOI: 10.16182/j.issn1004731x.joss.22-1550

参考文献:

[1] Wang Shuang, Guo Yanhe, Hua Wenqiang, et al. Semi-supervised PolSAR Image Classification Based on Improved Tri-training with a Minimum Spanning Tree[J]. IEEE Transactions on Geoscience and Remote Sensing, 2020, 58(12): 8583-8597.

[2] Li Zongyao, Togo R, Takahiro Ogawa, et al. Chronic Gastritis Classification Using Gastric X-ray Images with a Semi-supervised Learning Method Based on Tri-training[J]. Medical & Biological Engineering & Computing, 2020, 58(6): 1239-1250.

[3] Yin Chunyong, Cuan Haoqi, Zhu Yuhang, et al. Improved Fake Reviews Detection Model Based on Vertical Ensemble Tri-training and Active Learning[J]. ACM Transactions on Intelligent Systems and Technology, 2021, 12(3): 33.

[4] Khonde S R, Ulagamuthalvi V. Ensemble-based Semi-supervised Learning Approach for a Distributed Intrusion Detection System[J]. Journal of Cyber Security Technology, 2019, 3(3): 163-188.

[5] Zhao Jia, Li Song, Wu Runxiu, et al. Tri-training Algorithm Based on Cross Entropy and K-nearest Neighbors for Network Intrusion Detection[J]. KSII Transactions on Internet and Information Systems, 2022, 16(12): 3889-3903.

[6] 韩嵩, 韩秋弘. 半监督学习研究的述评[J]. 计算机工程与应用, 2020, 56(6): 19-27.
Han Song, Han Qiuhong. Review of Semi-supervised Learning Research[J]. Computer Engineering and Applications, 2020, 56(6): 19-27.

[7] Zhou Zhihua, Li Ming. Semi-supervised Learning by Disagreement[J]. Knowledge and Information Systems, 2010, 24(3): 415-439.

[8] 屠恩美, 杨杰. 半监督学习理论及其研究进展概述[J]. 上海交通大学学报, 2018, 52(10): 1280-1291.
Tu Enmei, Yang Jie. A Review of Semi-supervised Learning Theories and Recent Advances[J]. Journal of Shanghai Jiaotong University, 2018, 52(10): 1280-1291.

[9] 刘建伟, 刘媛, 罗雄麟. 半监督学习方法[J]. 计算机学报, 2015, 38(8): 1592-1617.
Liu Jianwei, Liu Yuan, Luo Xionglin. Semi-supervised Learning Methods[J]. Chinese Journal of Computers, 2015, 38(8): 1592-1617.

[10] 周志华. 基于分歧的半监督学习[J]. 自动化学报, 2013, 39(11): 1871-1878.
Zhou Zhihua. Disagreement-based Semi-supervised Learning[J]. Acta Automatica Sinica, 2013, 39(11): 1871-1878.

[11] Miller D J, Uyar H S. A Mixture of Experts Classifier with Learning Based on Both Labelled and Unlabelled Data[C]//Proceedings of the 9th International Conference on Neural Information Processing Systems. Cambridge, MA, USA: MIT Press, 1996: 571-577.

[12] Blum A, Chawla S. Learning from Labeled and Unlabeled Data Using Graph Mincuts[C]//Proceedings of the Eighteenth International Conference on Machine Learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001: 19-26.

[13] Chapelle O, Sindhwani V, Keerthi S S. Optimization Techniques for Semi-supervised Support Vector Machines [J]. The Journal of Machine Learning Research, 2008, 9: 203-233.

[14] Blum A, Mitchell T. Combining Labeled and Unlabeled Data with Co-training[C]//Proceedings of the Eleventh Annual Conference on Computational Learning Theory. New York, NY, USA: Association for Computing Machinery, 1998: 92-100.

[15] Zhou Zhihua, Li Ming. Tri-training: Exploiting Unlabeled Data Using Three Classifiers[J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(11): 1529-1541.

[16] 邓超, 郭茂祖. 基于自适应数据剪辑策略的Tri-training算法[J]. 计算机学报, 2007, 30(8): 1213-1226.
Deng Chao, Guo Maozu. ADE-tri-training: Tri-training with Adaptive Data Editing[J]. Chinese Journal of Computers, 2007, 30(8): 1213-1226.

[17] Li Dunming, Mao J, Shen Fuke. A Novel Semi-supervised Adaboost Technique Based on Improved Tri-training[C]//Proceedings of the 24th Australasian Conference on Information Security and Privacy. Cham: Springer International Publishing, 2019: 669-678.

[18] 张永, 陈蓉蓉, 张晶. 基于交叉熵的安全Tri-training算法[J]. 计算机研究与发展, 2021, 58(1): 60-69.
Zhang Yong, Chen Rongrong, Zhang Jing. Safe Tri-training Algorithm Based on Cross Entropy[J]. Journal of Computer Research and Development, 2021, 58(1): 60-69.

[19] Zhao Jia, Luo Yuhang, Xiao Renbin, et al. Tri-training Algorithm for Adaptive Nearest Neighbor Density Editing and Cross Entropy Evaluation[J]. Entropy, 2023, 25(3): 480.

[20] 王宇飞, 陈文. 基于DECORATE集成学习与置信度评估的 Tri-training 算法[J]. 计算机科学, 2022, 49(6): 127-133.
Wang Yufei, Chen Wen. Tri-training Algorithm Based on DECORATE Ensemble Learning and Credibility Assessment[J]. Computer Science, 2022, 49(6): 127-133.

[21] Gan Haitao, Sang Nong, Huang Rui, et al. Using Clustering Analysis to Improve Semi-supervised

Classification[J]. Neurocomputing, 2013, 101: 290-298.

[22] Wu Di, Shang Mingsheng, Luo Xin, et al. Self-training Semi-supervised Classification Based on Density Peaks of Data[J]. Neurocomputing, 2018, 275: 180-191.

[23] Alex Rodriguez, Alessandro Laio. Clustering by Fast Search and Find of Density Peaks[J]. Science, 2014, 344 (6191): 1492-1496.

[24] Zhao Jia, Wang Gang, Pan J S, et al. Density Peaks Clustering Algorithm Based on Fuzzy and Weighted Shared Neighbor for Uneven Density Datasets[J]. Pattern Recognition, 2023, 139: 109406.

[25] Angluin D, Laird P. Learning from Noisy Examples[J]. Machine Learning, 1988, 2(4): 343-370.

[26] Dua D, G`raff C. UCI Machine Learning Repository [EB/OL]. [2023-02-20]. https://archive.ics.uci.edu/.

[27] Fowlkes E B, Mallows C L. A Method for Comparing Two Hierarchical Clusterings[J]. Journal of the American Statistical Association, 1983, 78(383): 553-569.

[28] Janez Demšar. Statistical Comparisons of Classifiers Over Multiple Data Sets[J]. The Journal of Machine Learning Research, 2006, 7: 1-30.

[29] 贺朝, 康平, 李卿鹏, 等. 多策略集成萤火虫算法[J]. 南昌工程学院学报, 2023, 42(1): 80-87.
He Chao, Kang Ping, Li Qingpeng, et al. Firefly Algorithm with Combination of Multi-strategies[J]. Journal of Nanchang Institute of Technology, 2023, 42 (1): 80-87.

[30] Zhao Jia, Chen Dandan, Xiao Renbin, et al. Multi-strategy Ensemble Firefly Algorithm with Equilibrium of Convergence and Diversity[J]. Applied Soft Computing, 2022, 123: 108938.

DOI: 10.16182/j.issn1004731x.joss.22-1550