分类号 C93　　密级 公开
U D C　　编号 10741

兰州财经大学
LANZHOU UNIVERSITY OF FINANCE AND ECONOMICS

硕士学位论文

论文题目 密度峰值优化的分类算法研究

研究生姓名：刘学文
指导教师姓名、职称：聂飞平　教授
学科、专业名称：管理科学与工程
研究方向：信息管理与信息系统

提交日期：2022 年 5 月 29 日

独创性声明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：刘学文 签字日期：2022.5.29

导师签名：聂峰 签字日期：2022.5.29

关于论文使用授权的说明

本人完全了解学校关于保留、使用学位论文的各项规定，同意（选择“同意”/“不同意”）以下事项：

1.学校有权保留本论文的复印件和磁盘，允许论文被查阅和借阅，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文；

2.学校有权将本人的学位论文提交至清华大学“中国学术期刊（光盘版）电子杂志社”用于出版和编入 CNKI《中国知识资源总库》或其他同类数据库，传播本学位论文的全部或部分内容。

学位论文作者签名：刘学文 签字日期：2022.5.29

导师签名：聂峰 签字日期：2022.5.29

Research on Classification Algorithm with Density Peak Optimization

Candidate: Liu Xuewen

Supervisor: Nie Feiping

摘 要

密度峰值聚类(DPC)是一种新颖的聚类算法。DPC 能够利用数据的密度峰值信息发现潜在的类簇中心，基于样本之间的层次关系对任意形状数据快速分配标签。近年来，它在众多领域展示出了巨大的应用价值，并且越来越受到研究者重视。本文旨在研究密度峰值在半监督分类和不平衡数据分类领域的应用，主要工作如下：

（1）自训练算法的关键步骤是选取用于扩展训练集的高置信度样本。如果选取样本的伪标签不准确，会使训练的分类器的性能降低。为此，本文利用密度峰值隶属度筛选高置信度样本，提出了一种密度峰值隶属度优化的半监督自训练算法(STDPM)。首先，基于密度峰值定义了原型和近亲结点，以便更清晰地反映样本之间的层次关系。然后，提出了一种高置信度样本选取方法，先根据无标签样本与不同类簇内有标签样本之间的原型关系定义密度峰值类簇隶属度，从无标签近亲结点集中选取隶属度大于设定阈值的样本。最后选取样本由分类器赋予伪标签并用于扩展训练集。在 8 个公开的基准测试集上与 4 个算法进行对比实验，结果验证了高置信度样本选取方法的有效性。

（2）不平衡数据分类算法着重关注少数类样本的分类精度，但当多数类样本的损失过多信息时，整体分类精度会变低。为此，本文提出了一种密度峰值优化的球簇划分欠采样不平衡数据分类算法(DPBCPUSBoost)。首先，设计了一种保留高价值多数类样本的欠采样方法，利用密度峰值发现多数类中的代表性样本，并采用球簇划分方法找出决策边界区域内易误分的多数类样本，对这些样本赋予更高的采样权重，以尽量减少信息损失。然后，提出了一种融合类依赖和样本依赖的误分代价计算方法，根据样本类别分布信息计算不同类别样本的误分代价，根据密度峰值信息计算所有样本的误分代价，融合两种形式的代价作为样本的整体误分代价，新的误分代价计算方法充分考虑到了样本价值的差异性。最后，基于临时训练集训练分类器，并通过代价调整函数进一步增加高误分代价样本的权重。在 10 个 KEEL 基准测试集上与 4 个算法进行对比实验，结果验证了 DPBCPUSBoost 利用密度峰值优化欠采样方法和误分代价计算方法的有效性。

关键词：密度峰值 分类 自训练 不平衡数据分类 球 K 均值聚类

Abstract

Density Peaks Clustering (DPC) is a novel algorithm that uses the Density Peaks information of data to find potential cluster centers and quickly assigns labels to arbitrary-shaped data based on hierarchical relationships between samples. In recent years, DPC has shown great application value in many fields and has been paid more and more attention by researchers. This paper focuses on applying Density Peaks in Semi-Supervised Classification and Imbalanced Data Classification. The main work is as follows:

(1) It is the critical step to select high-confidence samples for expanding the training set in the Self-Training algorithm. It will degrade the performance of the trained classifier if pseudo-labels of selected samples are inaccurate. Therefore, we use Density Peaks Membership to select high-confidence samples and propose a Self-Training algorithm, named STDPM, for Density Peaks Membership optimization. Firstly, to more clearly reflect the hierarchical relationships between samples, we define Prototypes and Direct Relative Node base on Density Peaks. Secondly, we propose a method to select high-confidence samples. The method defines Density Peaks Membership according to the Prototype relationship between unlabeled samples and labeled samples in different clusters and then selects samples whose membership degree is greater than the set threshold from the set of Direct Relative Node. Finally, the classifier

gave the selected samples pseudo-labels, and we used them to expand the training set. We conducted comparative experiments on eight public benchmark test data sets, and the experimental results verify the method's effectiveness in selecting the high-confidence samples.

(2) The imbalanced data classification algorithm focuses on the classification accuracy of minority class samples. However, when the majority class samples lose too much information, the overall classification accuracy will be lower. Therefore, this paper proposes a Boosting algorithm of imbalanced data classification, named DPBCPUSBoost, based on Ball Cluster Partitioning and UnderSampling with Density Peaks optimization. Firstly, we design an undersampling method that retains the majority class samples with higher values. The method finds representative samples in the majority class cluster according to the Density Peaks information and takes the ball cluster division method to search for the majority class samples, which are easily misclassified in the decision boundary region. A higher sampling weight is assigned to those samples to reduce the loss of information. Secondly, we propose a misclassification cost calculation method that fuses class-dependent and sample-dependent. The method calculates the misclassification cost of different classes according to class distribution and calculates the misclassification cost of all samples according to Density Peaks information. Then, the method joins the two forms of cost as the overall misclassification cost of samples. The

new method of misclassification cost calculation fully considers the differences in the value of samples. Finally, we train a classifier on the temporary training set, and using the cost adjustment function further increases the weight of samples with high misclassification costs. We compared four algorithms with DPBCPUSBoost on ten KEEL benchmark test data sets. The experimental results verify the effectiveness of DPBCPUSBoost in optimizing the undersampling method and misclassification cost calculation method using Density Peaks.

Keywords: Density Peaks; Classification; Self-Training; Imbalanced Data Classification; Ball K-Means Clustering

目 录

1 引 言

1.1 研究背景与意义

随着计算机的微型化和传感器的普及化，人类从信息互联时代迈入万物互联时代。此时，数据规模呈爆发式增长，计算机的计算能力大幅提升，丰富的数据资源和强大的计算能力为机器学习的大规模应用创造了条件。当前，知识成为一种重要的生产要素，各个行业对知识的需求持续增加，机器学习作为一种可以自动从数据中获取知识的技术，越来越受到各行业的重视。

分类是机器学习中代表性最强、应用最广泛的监督学习(Supervised Learning，SL)[1,2]技术，它从有标签数据中学习出模型，由模型对新数据的类别进行预测。当前，虽然数据规模大幅增长，但是数据标签的获取代价十分高昂，大部分数据只含有少量标签，监督信息的不足严重限制了学习效果。聚类是一种无监督学习(Unsupervised Learning)技术，它可以在没有任何先验知识的情况下，根据数据的固有特征，将数据划分成有差异的子集。不同于监督学习和无监督学习方法，半监督学习方法(Semi-Supervised Learning, SSL)[3-5]允许我们从大量无标签数据和少量有标签数据中学习模型。半监督分类(Semi-Supervised Classification, SSC)是半监督学习[6]的一种代表性算法，它旨在利用无标签数据中的信息协助训练分类模型。自训练(Self-Training)是半监督分类算法家族中的一员，它是一种包裹算法，实现方式十分灵活，可以适应多样的半监督学习任务。自训练算法的核心步骤是从大量的无标签样本中筛选出少量的高置信度样本，由基分类器赋予这些样本伪标签，随后它们将被用于扩充训练集以增强基分类器的性能。然而，如果携带错误标签的样本也被添加进训练集，并且在迭代过程中传递，最终会使输出分类器的性能降低。为此，本文融合密度峰值聚类思想，提出了一种利用密度峰值隶属度选取高置信度样本的方法，该方法能有效提升自训练算法的性能。

传统分类问题假设数据分布平衡，然而在众多应用场景中，该假设难以成立。在信用欺诈检测和电信诈骗检测等场景，传统分类模型着重关注总体分类精度，在巨大的样本数量差距下，少数类容易被“忽视”，这类模型较难取得理想的效

果。在另一些场景中，少数类样本被误分的代价和多数类样本被误分的代价是不一致的。例如在对甲类传染病患者识别的场景中，将阳性患者判定为健康人的代价要远高于将健康人判定为患者的代价。近年来，各个行业对不平衡数据分类的需求不断增加，越来越多的研究者也将目光转向这个领域。为提高少数类样本的分类精度，研究者们提出了两种基本策略：一是数据重构，使多数类的样本数量减少或者使少数类的样本数量增加，从而实现数据分布平衡；二是在分类模型和思想上聚焦于少数类样本。上述两种策略殊途同归，都是通过给予更高价值的样本更多的关注，以此提升分类精度。本文融合两种策略，利用密度峰值信息发现潜在的类簇局部中心，利用球簇划分方法找出决策边界的“易误分区域”，通过欠采样和代价调整函数给予这些更高价值的样本更多的关注。实验结果表明，本文所提出方法可以有效提升不平衡数据分类性能，也为该领域研究提供了一种新的思路。

1.2 国内外研究现状

1.2.1 自训练算法研究现状

高质量数据获取难度大、成本高，通常情况下很多数据只含有少量标签，传统监督学习方法在数据标签受限的情况下效果不理想。与监督学习[1,2]和无监督学习[2]不一样，半监督学习[3-5]能够充分利用数据集内大量无标签数据和少量有标签数据中的信息来提升学习效果。自训练算法是一种经典的半监督学习算法框架，它的结构十分简单而且扩展性非常强，在众多应用场景中都表现出了良好效果，因此它受到了各界的青睐。如下从三个方面对自训练算法的研究现状进行介绍：

（1）训练数据扩展方式

自训练算法先基于由初始有标签样本构成的训练集(Training Set, *TS*)训练分类器，然后在迭代过程中向*TS* 添加带伪标签的样本获得扩展训练集(Extended Training Set, *ETS*)，高质量的*ETS* 可以增强分类器的性能，低质量的*ETS* 会弱化分类器性能。因此，采用何种训练数据扩展方式以获得更高质量的*ETS* ，是自训练算法中一个非常关键的问题。

Li 等人[7]提出了编辑自训练算法(Self-Training with editing, SETRED)，它采用割边权重统计(Cut Edge Weight Statistic, CEWS)[8,9]方法，通过左侧检验的方式，将含有较多割边的样本从训练集中移除。SETRED 的计算开销较大，而且当样本在相关邻近图(Relative Neighborhood Graph, RNG)上的近邻数量较少或者割边权重不平衡时，假设检验的效果不佳。聚类算法[10,11]可以从无标签样本中提取出有价值的信息，从而能够辅助选取高置信度样本。Gan 等人[12]基于聚类假设，提出了一种基于半监督模糊 C 均值聚类的自训练算法(Self-Training algorithm based on Semi-Supervised Fuzzy C-Means clustering, STSFCM)。STSFCM 在自训练迭代过程中嵌入了半监督模糊 C 均值聚类算法(Semi-Supervised Fuzzy C-Means algorithm, SSFCM[13,14])，将模糊类簇隶属度大于设定阈值的样本作为高置信度样本。STSFCM 从训练集中移除类簇归属确定性低的样本是一种有效的思路，然而该算法对非球形数据聚类的效果不理想，且易受初始点选择的影响。因此，有研究者尝试在自训练算法中嵌入密度峰值聚类算法(Clustering by fast search and find of Density Peaks, DPC)[15]。DPC 可将任意维的数据映射成二维，利用密度峰值在二维空间内建构数据之间的层次关系[16]，而且它对非球形数据也具有较好的适应性。近年来，有关 DPC 的研究热度不减，文献[17-22]从密度峰值计算、聚类中心选取和数据点分配等方面提出了改进方法，文献[23,24]利用密度峰值对 K 中心点算法(K-Medoids)、模糊 C 均值聚类算法(Fuzzy C-Means, FCM)中初始聚类中心、类簇数目选择等过程进行了优化。Wu 等人[25]提出基于数据密度峰值的自训练算法(Self-Training based on Density Peaks of data, STDP)，基于数据间的潜在层次关系，先标记有标签样本的无标签“前驱”，再标记无标签“后继”，该方法扩展训练集的速度较快。文献[26]通过局部密度阈值将样本划分为核心点和边界点，并依次赋予标签、添加进训练集。卫丹妮等人[27,28]将 STDP 与 SETRED 结合，用 CEWS 方法再次筛选高置信度样本，进一步提升了训练集的质量。

（2）基分类器应用方式

自训练算法中 *ETS* 内样本的伪标签由基分类器分配，而基分类器通常由一个或多个分类器构成[29,30]，不同的分类器及其组合方式会产生不同的效果。因此，选择合适的分类器及其组合方式，对提升算法的整体分类性能是至关重要的。

Gan 等人[12]通过模糊隶属度阈值筛选出高置信度样本，然后再由 SVM 分配

伪标签。Karlos 等人[31]通过测试样本的 K 近邻(K Nearest Neighbors)构建局部朴素贝叶斯模型(Naive Bayes, NB)并输出后验概率，将大于设定阈值的样本作为高置信度样本。单个分类器构成的自训练算法通常结构比较简单、容易实现，而且它的整体复杂度不高。文献[32]和文献[33]同时训练多个分类器并输出多个后验概率值，取平均值作为无标签样本的置信度，Shi 等人[34]则通过投票的方式选取高置信度样本。多个分类器组合的方式提升性能，但是也提高了算法的复杂度，采取哪一种基分类器应用方式需要根据具体的应用场景来决定。

（3）迭代训练停止方式

自训练算法通过不断迭代训练输出一个更强的分类器，何时停止迭代直接关系到所输出分类器的性能。如果迭代停止过早，分类器可能训练不充分，如果停止过晚，*ETS* 内一些被误标记的样本可能会被添加到训练集中，从而导致分类器性能不断降低。因此，如何停止迭代训练过程也是自训练算法中需要着重关注的问题。

SETRED[7]在每轮迭代时，均从无标签样本集中随机抽取部分样本训练分类器，在达到固定次数后停止迭代。STSFCM[12]通过阈值来控制迭代过程，当无标签样本数量小于或等于设定阈值时算法将停止迭代。STDP[25]在所有无标签样本都被赋予标签后，停止迭代并输出分类器，文献[35]则在此基础上增加了新条件，即分类器稳定之后迭代训练过程可以终止。通过设定固定阈值或固定迭代次数停止迭代的方式适应性更强，针对不同的数据集可以设置不同的停止参数，但是需要额外的调参步骤。而当分类器稳定或无标签样本集稳定后停止的方式具有无参的优点，但该方式牺牲了一些灵活性。

1.2.2 不平衡数据分类算法研究现状

在一些应用场景如异常检测[36-39]、信用欺诈检测[40-42]、医保欺诈检测[43]、电信诈骗检测[44,45]等，数据分布不平衡现象广泛存在，即数据集内不同类别样本分布不平衡或者同一类别样本内部分布不平衡。因为传统分类算法对所有样本“一视同仁”，导致一些价值较高但是样本数量较少的类别被“忽视”。因此，研究者们提出了两种基本策略来增加对少数类的“关注度”：一是对数据进行重构，对

多数类样本或者少数类样本数量进行调整，使不同类别的样本数量差距缩小；二是在分类模型[46]和思想上聚焦于少数类的分类精度[47]。

（1）数据重构策略

数据重构策略最常用的方法是欠采样[48,49]、过采样[50-52]以及融合欠采样和过采样的混合采样[53,54]。过采样方法与欠采样方法有明显区别，前者增加少数类样本数量，后者则减少多数类样本，但都是以平衡样本分布为目的。通常，较大的数据集宜采用欠采样方法，因为采用过采样方法会使数据量大大增加，从而大大提高训练时间。随机欠采样方法(Random Under Sampling, RUS)[48]是最简单的欠采样方法之一，其使用完全随机的方法移除多数类样本。显然，RUS 方法并没有考虑到多数类样本所携带的信息，极可能将一些对后续分类过程有价值的样本移除。Lin 等人[55-57]融合聚类思想，提出了基于聚类的欠采样方法。聚类方法能够发现多数类样本中那些具有代表性的样本，在欠采样过程保留这些样本，可以尽量避免信息损失。

（2）分类模型及思想改进策略

分类模型改进策略旨在重新设计分类模型以增加对少数类的关注度，分类思想改进策略不对基础分类模型进行改进，而是采用不同的思想使分类器增加对少数类的关注度[47]，典型的分类思想包括集成学习[58]、代价敏感学习[59]、单类学习[60]和主动学习[61]等。Liu 等人[62]对 C4.5 改进，提出了一种鲁棒的决策树 CCPDT(Class Confidence Proportion Decision Tree)，它用 CCP(Class Confidence Proportion)替换原有的置信度，减弱了数据分布不平衡造成的影响。Fan 等人[59]将集成学习和代价敏感学习结合，提出了代价敏感自适应增强算法(Cost-sensitive AdaBoost, AdaCost)。自适应增强(Adjusts Adaptive Boosting， AdaBoost)[58]赋予每个样本同等的误分代价，AdaCost 则对少数类样本赋予更高的误分代价，以增加对少数类样本的关注度。

（3）混合策略

现有研究中，不同策略相结合的效果可能比单一策略更好。Seiffert 等人[63]基于 AdaBoost 提出随机欠采样增强方法(Random UnderSampling Boosting, RUSBoost)，该方法在迭代过程中对多数类进行随机欠采样，并与少数类组成临时训练集，基于临时训练集和样本权值训练弱分类器。RUSBoost 采用 RUS 方法

平衡数据的分布，而当不平衡比例非常高时，可能需要非常多的迭代次数，才能弥补随机采样造成的信息损失。Liu 等人[64]将 RUS 方法和集成学习方法相结合，提出了 EasyEnsemble 和 BalanceCascade 方法。EasyEnsemble 方法从多数类样本中随机抽取与少数类样本数量相等的多个子集，分别与少数类样本组合成新训练集，独立训练多个 AdaBoost[58]分类器，最终输出集成分类器。BalanceCascade 与 EasyEnsemble 不同之处在于，每次迭代时，利用分类阈值从训练集中移除分类正确的样本。上述两种方法相较于 RUS 方法，减少了多数类样本的信息损失，但是显著增加了训练时间。王俊红等人[49]采用 AdaCost[14]的样本权重更新策略，提出了基于欠采样和代价敏感的不平衡数据分类算法(UnderSamples and Cost-sensitive Boosting, USCBoost)。USCBoost 在初次迭代时仍采用 RUS 方法，但在后续迭代时只对权重较小的样本进行欠采样。此外，USCBoost 采用了 AdaCost 的代价调整函数，为少数类样本定义比多数类样本更高的误分代价，但是它没有考虑到同一个类别内样本之间也存在差异[24]。

1.3 研究内容和创新点

DPC 是 Rodriguez 等人在 Science 上提出的一种聚类算法，它利用密度峰值发现类簇的局部中心，基于密度峰值信息构建样本之间的层次关系，然后利用这种特殊的层次关系实现快速聚类。本文对 DPC 进行深入研究后，发现密度峰值和基于密度峰值构造的层次关系可以用于解决目前自训练算法和不平衡数据分类算法存在的一些问题。本文的主要研究内容和创新点如下：

（1）密度峰值隶属度优化的自训练算法

自训练算法的关键在于高置信度样本选取，本文利用密度峰值优化高置信度样本选取过程，主要创新如下：基于密度峰值，提出了两个描述样本之间层次关系的定义——原型以及原型树中样本的近亲结点；根据无标签样本与有标签样本在类簇中的分布情况，定义了“密度峰值隶属度”，新的类簇隶属度充分利用了数据的结构信息，能够适应非球形数据；提出了一种高置信度样本选取方法，从有标签样本的无标签近亲结点集中，将密度峰值隶属度较大的样本作为高置信度样本，提出的高置信度样本选取方法具有较高的运行效率，通过该方法扩展训练

集能够实现较好的分类性能。

（2）密度峰值优化的球簇划分欠采样不平衡数据分类算法

不平衡数据分类算法需要给予更高价值样本更多的关注，本文利用密度峰值和球簇划分发现高价值的多数类样本，并通过欠采样和代价调整给予高价值样本更多的关注，主要创新如下：提出了一种密度峰值优化的球簇划分欠采样方法，通过密度峰值发现多数类中更具有代表性的局部中心点，赋予它们更高的采样权重，然后利用球簇划分找出决策边界区域内的样本，增大它们的采样权重，最后根据采样权重对多数类样本进行欠采样，提出的欠采样方法尽量避免了多数类样本的信息损失；提出了一种兼顾类依赖和样本依赖的误分代价计算方法，对少数类样本赋予更高的误分代价的同时，根据样本在类簇中的代表性，赋予不同样本不同的误分代价，新的误分代价计算方式能够更准确的反映样本之间价值的差异性。

1.4 本文组织结构

本文利用密度峰值对自训练和不平衡数据分类算法进行改进，分别提出了一种算法。全文的组织结构如下：

第 1 章为引言，简要介绍了本文的研究背景和意义，介绍了自训练和不平衡数据分类算法的研究现状，最后简要介绍了本文的研究内容和组织结构。

第 2 章介绍了与本文研究相关的一些基础理论知识。首先介绍了一些常用的分类和聚类算法，然后介绍了 USCBoost 算法、DPC 算法和 Ball K-Means 算法的主要思想，最后对半监督学习的核心理论做了简要介绍。

第 3 章对本文提出的密度峰值隶属度优化的自训练算法进行了详细的介绍，并在 8 个基准数据集上与 4 个相关算法进行对比实验，以验证提出算法的有效性。

第 4 章对本文提出的密度峰值优化的球簇划分欠采样不平衡数据分类算法进行了详细的介绍，在 10 个基准数据集上与 4 个相关算法进行对比实验，以验证提出算法的有效性。

第 5 章对本文的研究工作进行了总结，并对未来的研究工作进行了展望。

2 相关理论介绍

2.1 分类学习

分类是机器学习领域的主要研究方向之一，它从训练数据中学习出分类模型，然后使用该模型对新输入待预测数据的类别进行预测。分类学习过程的形式化描述为：对于给定的训练数据集$\boldsymbol{T}=\{(\boldsymbol{X},\boldsymbol{Y})\}=\{(x_1,y_1),(x_2,y_2),\cdots,(x_t,y_t)\}$和预测数据集$\boldsymbol{P}=\{x_{t+1},x_{t+2},\cdots,x_{t+p}\}$，其中$x_i\in\mathbb{R}^{1\times d}$为一个样本点，$d$为$x_i$的维度，$y_i$为$x_i$的类别标签，$t$和$p$分别为训练集和预测集的样本数量。首先在$\boldsymbol{T}$中学习一个从特征空间$\boldsymbol{X}$到标签空间$\boldsymbol{Y}$的映射函数$f:\boldsymbol{X}\rightarrow\boldsymbol{Y}$，然后由$f$对$\boldsymbol{P}$内样本点的类别进行预测，即通过函数$f$为每一个$x_i\in\boldsymbol{P}$都分配了标签$y_i$。

本节简要介绍 4 种基本分类问题、4 种经典分类算法、1 种不平衡数据分类算法以及多个分类性能评价指标。

2.1.1 分类基本问题

分类算法有特定的适用场景，不同类型的分类问题需要选择不同的分类算法，基本的分类问题类型有以下四种：

（1）二类别数据分类

通常在电信诈骗检测、垃圾邮件检测、信用欺诈检测等机器学习应用场景，数据标签只包括两个类别，类似这样只针对两类别数据进行建模预测的任务称为二分类(Binary Classification)，这两个类别通常表示“是非”或“有无”的含义，其类别标签值$y_i=\{0,1\}$或者$y_i=\{-1,+1\}$。二分类是最简单的分类问题，绝大多数分类算法都可以用于二分类，少部分算法只能用于二分类，例如逻辑回归(Logistics Regression, LR)算法、经典支持向量机(Support Vector Machines, SVM)算法等。

（2）多类别数据分类

多类别数据分类又称为多分类(Multi-Class Classification)，它是最常见的一种

分类问题。多类别数据是指数据集中每个样本只属于一个类并且数据集的类别数量多于两个，其形式化描述为 $y_i=\{class(j)||class|\geq 2, j=1,2,\cdots,k\}$，其中 k 是类别数量，$class(j)$ 表示第 j 个类别的标签。

（3）多标签数据分类

多标签数据是指同一个样本所属的类别可能有一个或多个，多标签数据分类(Multi-Label Classification)与多分类是两个不同类型的问题，大部分多分类算法不适用于多标签数据。通常解决多标签数据分类问题可以采用问题转换策略(Problem Transformation Methods)将多标签分类问题转化为二分类或多分类问题，或者采用一些专门针对多标签数据的分类算法，例如多标签决策树(Multi-Label Decision Tree, ML-DT[65])、多标签 K 近邻(Multi-Label K-Nearest Neighbor, ML-KNN[66])、多标签支持向量机(Ranking Support Vector Machine, Rank-SVM[67])等。

（4）不平衡数据分类

数据不平衡是指数据集中不同类别样本的数量相差较大即类间分布不平衡，或者同一类别内部样本数量差异较大即类内分布不平衡。对于分布不平衡的数据，传统算法往往过于关注多数类样本，从而导致少数类样本的分类效果不理想。因此，解决不平衡数据分类问题，可以通过数据重构以增加少数类样本或减少多数类样本从而使数据分布趋于平衡，以间接的方式增加对少数类样本的关注度，代表方法有和 SMOTE[68]，或者对分类算法和思想进行改进，用更加直接的方式增加对数量不占优势的样本的关注度，代表方法有 RUSBoost 和 SMOTEBoost[69]。

2.1.2 经典分类算法介绍

本节主要介绍 4 种与本文研究密切相关并且具有很强代表性的分类算法：决策树(Decision Tree)，K 近邻(K Nearest Neighbors, KNN)，支持向量机(Support Vector Machine, SVM)和自适应增强(Adaptive Boosting, AdaBoost)。

（1）Decision Tree

Decision Tree 是机器学习中最常用的一种预测技术之一，它具有解释性强、运行速度快和可处理标称型数据等众多优点。Decision Tree 的核心步骤如下：

（a）特征选择：从数据的全部特征中筛选出一个特征作为当前结点的分裂

准则，不同的衍生算法会采用不同的准则。

（b）树生成：根据所筛选出的特征，自上而下地递归生成子结点直至特征无法划分停止。

（c）树剪枝：当树分支过多时易出现过拟合和解释能力下降的现象，需要对分支进行修剪以降低过拟合程度、增强可解释性。

Decision Tree 从概念学习算法(Concept Learning System, CLS)发展而来，由 ID3 算法发扬光大，随后涌现出大量衍生算法，不同算法在特征选择、树生成、树剪枝、缺失值处理和数据不平衡处理等步骤上存在差异。例如 ID3 算法无法处理缺失值也不能进行剪枝操作，而 C4.5 和 CART 则能够进行缺失值处理和剪枝操作。

（2）KNN

KNN 是一种基于实例的分类算法，其核心思想是一个实例的类别应与 K 个最相似实例中多数类的类别一致。KNN 的基本步骤是找出待预测样本的 K 个最近邻，计算这 K 个最近邻的类别分布，根据类别分布投票决定待预测样本的类别。KNN 是一种没有显示学习过程的惰性学习算法，其实现非常简单、参数少、既能用于分类也能用于回归，但是空间复杂度较高、占用内存量大、对不平衡数据的适应性较差。

（3）SVM

SVM 是一种非线性分类算法，它根据结构风险最小化准则，采用最大化分类间隔的策略构造最优分类超平面，构造过程可形式化为求解凸二次规划的问题。对于分类问题，SVM 根据区域中的样本计算该区域的决策曲面，由此确定该区域中未知样本的类别。SVM 具有适用高维数据、泛化能力强和非线性等优点，但是也存在复杂度高、解释能力弱、只支持二分类等缺点，因此研究人员又提出了很多改进算法。Platt[70]提出序列最小优化算法(Sequential Minimal Optimization, SMO)，以减少大规模样本求解过程中的计算量。台湾大学林智仁博士开发了一个评价很高的通用工具包 LIBSVM①，它能够用于解决回归、多分类、不平衡数据分类等问题。

（4）AdaBoost

① https://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html

AdaBoost是一种经典的Boosting算法，其核心思想是训练一系列弱分类器，并根据准确度将它们加权组合为一个更强的分类器，在训练过程中不断增加被误分样本的权重来提高对这些样本的识别率[71]。AdaBoost 的基本步骤如下：

（a）将训练数据的样本权重初始化为$\frac{1}{n}$，其中n为数据集中样本的数量。

（b）训练一个弱分类器，计算该分类器的误差率和它在最终分类器中的权重，并使误差率小的分类器的权重更大。

（c）更新样本的权重，使被误分的样本的权重增加，被正确分类的样本的权重降低。

（d）满足迭代停止条件则输出弱分类器加权组合后的强分类器。

AdaBoost 具有精度高、扩展性强和不易过拟合等优点，但是它的训练时间较长，且容易受噪声影响。AdaBoost 作为一种算法框架，它的性能依赖于弱分类器的选择。

2.1.3 基于欠采样和代价敏感的不平衡数据分类算法

RUSBoost 是一种融合欠采样和集成学习的不平衡数据分类算法，它在AdaBoost迭代训练分类器之前，采用RUS方法对多数类样本进行欠采样，与少数类样本合成分布较为平衡的临时训练集。

AdaCost是一种融合集成学习和代价敏感学习的不平衡数据分类算法，它在AdaBoost基础上增加了一个代价调整函数$\beta(sign(y_i h_t(x_i)), c_i)$，其中$h_t$是第$t$次迭代训练的分类器，$c_i$为样本$x_i$的误分代价。用$\beta_+$表示分类器输出结果与真实标签一致即$sign(y_i h_t(x_i))$等于 1 时的调整函数，此时$\beta_+(c_i) = -0.5c_i + 0.5$，用$\beta_-$表示分类器预测错误时的调整函数，此时$\beta_-(c_i) = 0.5c_i + 0.5$。

王俊红等人[49]基于RUSBoost和AdaCost提出了USCBoost算法。USCBoost在首轮迭代时对多数类进行随机欠采样，后续迭代过程中只选取权值较高的多数类样本参与训练分类器，并通过代价调整函数增加高误分代价样本的权重。

给定训练样本集$X = \{x_1, x_2, \cdots, x_n\}$，标签集$Y = \{y_1, y_2, \cdots, y_n\}$，令$X_{\text{maj}}$表示多

数类样本，$X_{\min}$ 表示少数类样本，则 USCBoost 算法的步骤如表 2.1 所示：

表 2.1 USCBoost 算法步骤

输入：	训练样本集 X，标签集 Y，迭代次数 T，弱分类器 h；
输出：	集成分类器 H；
初始化：	样本分布权重：$W_1=\{w_{11},w_{12},\cdots,w_{1n}\}$，$w_{1i}=1/n$，$i=1,2,\cdots,n$。
步骤 1	计算误分代价：$c_i(y_i=-1)=\mid X_{\min}\mid/n$，$c_i(y_i=1)=\mid X_{\mathrm{maj}}\mid/n$；

步骤 2	For $t\in[1,T]$ Do:
	If $t=1$:
	根据采样率$\mid X_{\min}\mid/\mid X_{\mathrm{maj}}\mid$对多数类样本进行随机欠采样；
	Else:
	选取多数类样本中权重较大的前$\mid X_{\min}\mid$个样本；
	End If
	欠采样后的多数类样本与全部少数类样本组成临时训练集 D_t，归一化 W_t，在 D_t 上根据 W_t 训练分类器 $h_t(x)$；
	计算分类误差：$e_t=\sum_{i=1}^{n}w_{ti}I(h_t(x_i)\neq y_i)$；
	计算分类器权重：$\alpha_t=\frac{1}{2}\ln((1-e_t)/e_t)$；
	计算权重因子：$\beta_i=-\frac{1}{2}y_ih_t(x_i)c_i+\frac{1}{2}$；
	计算样本权重：$W_{t+1}=W_t(i)\exp(-\alpha_ty_ih_t(x_i)\beta_i)/Z_t$，$Z_t$ 为归一化因子；
	End For

步骤 3	输出组合的分类器 $H(x)=\operatorname{sign}\sum_{t=1}^{T}\alpha_th_t(x_i)$。

2.1.4 分类性能评价指标

评价指标是指在固定输入数据的情况下，对不同算法或者同一算法不同参数性能优劣进行度量的定量指标。每一种评价指标都有其侧重点，只能反映模型在特定方面的性能，因此，想要客观评价某一算法的性能需要综合多个评价指标。

本节介绍一些常用的分类性能评价指标：准确率(Accuracy)、精确率

(Precision)、召回率(Recall)/灵敏度(Sensitivity)/真正例率(True Positive Rate, TPR)、假正例率(False Positive Rate, FPR)、特异度(Specificity)/真负例率(True Negative Rate, TNR)、F1 分数(F1-Score)、几何均值(G-mean)等。

在二分类算法中，假设样本的类别分为正例和负例，则待预测样本的类别存在以下四种情况：

（1）实际是正例却被预测为正例，该样本用 TP(True Positive)表示

（2）实际是负例却被预测为正例，该样本用 FP(False Positive)表示

（3）实际是负例却被预测为负例，该样本用 TN(True Negative)表示

（4）实际是正例却被预测为负例，该样本用 FN(False Negative)表示。

由以上四种分类预测情况，可以绘制二分类的混淆矩阵(Confuse Matrix)。如表 2.2 所示,列表示预测类别，行表示实际类别。

表 2.2 二分类混淆矩阵

预测类别 / 实际类别	正例	负例
正例	TP	FN
负例	FT	TN

由混淆矩阵我们可以很方便地对多个评价指标进行计算：

（1）准确率：正确预测的样本数占总样本数的比值。

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2-1}$$

（2）精确率：正确预测的正例样本数占被预测为正例样本数的比值。

$$Precision = \frac{TP}{TP + FP} \tag{2-2}$$

（3）召回率（真正例率）：正确预测的正例样本数占总正例样本数的比值。

$$Recall = TPR = \frac{TP}{TP + FN} \tag{2-3}$$

（4）特异度（真负例率）：正确预测的负例样本数占总负例样本数的比值。

$$Specificity = \frac{TN}{TN + FP} \tag{2-4}$$

（5）假正例率：错误预测的正例样本数占总负例样本数的比值。

$$FPR = \frac{FP}{FP + TN} \tag{2-5}$$

（6）F1 分数：精确率和召回率的调和均值。

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2-6}$$

（7）几何均值：灵敏度和特异度的几何均值。

$$G - mean = \sqrt{Sensitivity * Specificity} \tag{2-7}$$

（8）AUC：ROC 曲线下的面积。

ROC(Received Operating Characteristic)曲线也称作接受者操作特征曲线，它是以 FPR 为横坐标，TPR 为纵坐标绘制出来的曲线，曲线下的面积(Area Under Curve, AUC)越大，通常能够表明分类器的性能越好，而且 ROC 曲线受正负样本分布的影响较小，它可以较客观地评价分类模型的性能。

2.2 聚类学习

聚类是一种非常典型的无监督学习算法，它不依赖先验知识而是根据输入数据中的潜在信息划分类簇，尽量使同一簇内数据之间的差异较小、不同簇内数据之间的差异较大。聚类过程可以形式化描述为：对于待划分的数据集 $X = \{x_1, x_2, \cdots, x_n\}$ 及类别数量 k，其中 $x_i \in \mathbb{R}^{1 \times d}$ 为一个样本点，d 为 x_i 的维度，n 为样本数量。根据选择的相似性(Similarity)度量方式，将 X 划分为不相交的若干个子集 $\{X_1, X_2, \cdots, X_k\}$，对任意的 $l \neq m$，最大化 $Similarity(X_l)$ 及最小化 $Similarity(X_l, X_m)$，$X_l \cap X_m = \varnothing$ 且 $X = X_1 \cup X_2 \cup \cdots \cup X_k$，最后对 X_l 内的样本分配一个标签 Y_l，$Y_l = \{class(j) \mid j = 1, 2, \cdots, k\}$，其中 $class(j)$ 为第 j 个类别的标签。

本节主要介绍五种常见的聚类算法类型和两种比较新颖的聚类算法——密度峰值聚类算法(DPC)和球 K 均值聚类算法(Ball K-Means)。

2.2.1 聚类算法类型

聚类算法种类繁多，根据算法的特点，大致可以分为五类：基于层次的聚类、基于划分的聚类、基于密度的聚类、基于网格的聚类和基于模型的聚类。

（1）基于层次的聚类

层次聚类算法(Hierarchical Clustering Method)的核心思想是基于发现的树形层次结构，将数据划分到不同层的簇。它的优点是无需事先指定需要聚成的类别数目，在树形结构构造完成后可以根据实际需要任意设定。因聚类过程的差异，层次聚类又可以分为自顶向下(Top-Down)的分裂聚类和自底向上(Bottom-Up)的合并聚类。自顶向下的分裂聚类首先将全部样本作为一个簇，然后从根结点开始递归地分裂出更小的子簇，直至满足终止条件停止分裂。自顶向下的合并聚类首先将每个样本都作为一个簇，然后根据一定的准则，逐步将最相似的子簇合并，直至满足终止条件停止合并。

常用的层次聚类算法包括平衡迭代削减聚类算法(Balanced Iterative Reducing and Clustering using Hierarchies, BRICH[72])、鲁棒的分类属性聚类算法(Robust Clustering using Links, ROCK[73])、代表点聚类算法(Clustering using Representatives, CURE)[74]和变色龙聚类算法(A Hierarchical Clustering Algorithm using Dynamic Modeling, Chameleon)[75]。BRICH 算法适宜用于大规模数值型数据聚类，其利用树结构对数据进行划分，将叶结点和低层次结点作为子簇，然后借助其他聚类方法将这些子簇聚成更大的簇。ROCK 算法主要用于标称型数据聚类，其根据子簇之间的互联相似度进行簇合并。CURE 算法在每个簇中选取一些代表点，使其能够适应非球形数据和不平衡数据，而且大大提高了处理效率、降低了噪声敏感度。Chameleon 算法首先构造 KNN 图并分割为多个子图（子簇），然后同时考虑簇间的互联性和近似性，根据互联度和近似度的乘积，每次都将具有最大乘积值的两个子簇合并。Chameleon 算法针对 ROCK 和 CURE 算法进行了改进，通过动态建模计算簇间的相似度，能够适应各种复杂形状的数据。

（2）基于划分的聚类

划分聚类算法旨在实现“簇内相似度最大化，簇间相似度最小化”的效果。划分聚类算法的主要步骤是：首先根据用户设定的类簇数k，选取k个初始样本点作为质心，然后不断迭代重置质心，不断优化目标函数，最后在目标函数达到最优时结束得到k个簇。

划分聚类算法的代表性算法有 K-Means 系列算法、迭代自组织聚类算法(Iterative Self-organizing Data Analysis Techniques Algorithm, ISODATA)和 FCM 算

法等。K-Means 是最具代表性的聚类算法，它将类簇的虚拟质心作为参照点，根据样本与质心的相似度来划分所属的类簇。K-Means 具有原理简单、复杂度低的优点，但存在对噪声敏感、对初始点选择敏感以及不适用于非球形数据等问题。针对 K-Means 存在的问题，研究人员提出了很多改进算法。K-Means++在 K-Means 基础上，使选择的初始质心相距更远，大大提高了算法效率。对于超大规模的数据，类簇数量往往难以确定，ISODATA 通过删除样本数较小的类别、拆分样本数较多的类别，能够自动选择恰当的类簇数量。K-Medoids 通过选取真实存在的样本点作为类簇中心，一定程度上降低了噪声的干扰。Kernel K-means 借助核函数，将样本映射到另一个特征空间进行聚类，解决了 K-Means 不适用于非球形数据的问题。模糊 C 均值聚类（Fuzzy C-Means，FCM）融合模糊理论对 K-Means 进行了推广，与 K-Means 只能输出样本确定的类别不同，FCM 能输出 [0,1] 范围内的类簇隶属度。

（3）基于密度的聚类

密度聚类(Density-Based Clustering)的核心思想是将密度大于设定阈值的样本点分配到最近的簇，基于样本之间的可连接性逐步扩展簇。密度聚类算法具有对噪声数据不敏感、适用于非球形数据等优点，但是其复杂度较高，处理大规模数据的效率不高。

密度聚类的代表算法有：DBSCAN 算法(Density-Based Spatial Clustering of Applications with Noise)、OPTICS 算法(Ordering Points To Identify Clustering Structure)、DENCLUE 算法(DENsity based CLUstEring)和 MDCA 算法(Maximum Density Clustering Algorithm)。DBSCAN 基于邻域搜索将样本点划分成核心点、边界点和噪声点，从核心点出发将密度相连的点合并成一个簇。DBSCAN 能够发现任意形状的簇，且无需先设定类簇数即可完成聚类。OPTICS 对 DBSCAN 的邻域半径依赖问题进行了改进，它根据给定的邻域半径和最小邻近点生成有序列表决策图，通过可达距离阈值交互式地输出聚类结果。DENCLUE 使用影响函数(Influence Function)来描述样本点在邻域内的影响，用关联点的“影响函数和”来描述样本点的全局密度，全局密度函数的极大值作为密度吸引点，将大于密度阈值的密度吸引点的关联样本划分到同一个簇，小于密度阈值的密度吸引点的关联

样本则作为噪声。MDCA 将密度聚类思想引入到划分聚类算法中，利用密度阈值确定样本所属的基本簇，根据最大密度点与其邻近点的距离不断扩展基本簇。

（4）基于网格的聚类

网格聚类算法(Grid-Based methods)根据样本属性取值将样本空间划分成网格，每个样本都映射到网格单元中，由密度阈值判断网格单元是否稠密，将相邻的稠密单元合并为一个类。网格聚类算法的复杂度取决于网格每一维的单元数量，数据维度增加将使算法运行效率大幅下降，因此网格聚类算法不适用于高维数据，但是算法在低维大规模数据上表现良好。

网格聚类算法的代表性算法有：统计信息网格算法(STatistical INformation Grid, STING)、子空间聚类算法(CLustering In QUEst, CLIQUE)、小波聚类算法(Wave-Cluster)。STING 是一种基于网格的多分辨率聚类算法，它根据样本特征的统计信息将样本空间递归地划分为多层网格，高层级网格单元被细分为多个低层级单元，从查询层级向下逐层查找相关性高的单元，到达最底层时对相关单元进行分组。CLIQUE 是一种结合了密度聚类思想的子空间聚类算法，它将样本空间划分成多维的网格，在每个维度上搜索邻近的稠密网格，遍历网格并将稠密网格及其邻近稠密网格合并成一个类。Wave-Cluster 是一种多分辨率聚类算法，它通过多维网格结构统计样本信息，利用小波变换将原有样本空间变换为频域空间，并在新的空间中搜索稠密网格。Wave-Cluster 的时间复杂度低，适合处理低维大规模数据，而且由于小波变换的多分辨率特性，它能进行不同精确度的聚类。

（5）基于模型的聚类

基于模型的聚类算法的核心思想是给每个类簇预设一个模型，然后在样本空间中找出对该模型拟合最好的样本。基于模型的聚类算法主要包括基于概率模型的算法和基于神经网络模型的算法。

基于概率模型的聚类算法是指概率生成式模型(Generative Model)，其假设同类样本均服从同一概率分布。常用的基于概率模型的算法是高斯混合模型（Gaussian Mixture Models, GMM）。GMM 是一种软聚类(Soft Clustering)算法，它假定全部样本均服从一个高斯混合分布，每个类簇的样本各自服从一个高斯分布，利用期望最大化(Expectation Maximization, EM)法来学习高斯分布中待定的参数，并由确定的概率分布模型来估计样本点隶属于不同类簇的概率。

常用的基于神经网络模型的聚类算法是自组织映射算法（Self Organizing Maps, SOM）。SOM 将每个样本映射到与其最近的竞争层神经元上，在竞争层上生成一个低维、离散的映射,尽可能地保持了拓扑结构和样本间的相对距离，经过竞争、协作和适应这三个学习过程，可以输出样本隶属于不同类簇的概率。

2.2.2 密度峰值聚类算法

Rodriguez 等人[15]在 Science 上提出了密度峰值聚类算法(DPC)，其基本思想是：类簇中心与比它密度更高的样本点相距较远，类簇中心被更低密度的样本点所围绕。根据上述思想，Rodriguez 等人提出了“局部密度”和“峰值”概念。

样本 x_i 的局部密度（高斯核） ρ_i 为：

$$\rho_i = \sum_{j \neq i} e^{-(\frac{d_{ij}}{d_c})^2} \tag{2-8}$$

其中 $d_c = \hat{d}_{m \cdot \alpha\%}$ ， $\hat{d}$ 表示距离序列 $\left\{d_{12}, d_{13}, \cdots, d_{23}, \cdots, d_{(n-1)n}\right\}$ 的升序排列 $\left\{\hat{d}_1, \hat{d}_2, \cdots, \hat{d}_m\right\}$ ， $\hat{d}_1 < \hat{d}_2 <, \cdots, < \hat{d}_m$ ， n 表示样本的个数， m 表示距离序列的个数， α 表示截断距离截取比例阈值。

样本 x_i 的峰值 δ_i 为：

$$\delta_i = \begin{cases} \min(d_{ij}) & \forall j \neq i, \rho_i < \rho_j \\ \max(d_{ij}) & \forall j \neq i, \rho_i > \rho_j \end{cases} \tag{2-9}$$

由式(2-9)知，样本的峰值为与“密度高于它且离它最近”的样本之间的距离。

样本 x_i 的密度与峰值的乘积 γ_i 为：

$$\gamma_i = \rho_i \cdot \delta_i \tag{2-10}$$

由 DPC 的基本思想可知， ρ_i 与 δ_i 越大，则样本 x_i 成为类簇中心的概率也越大。因此，DPC 通过一个决策图来选出 ρ_i 与 δ_i 都较大的样本，并将这些样本作为类簇中心。在选取出类簇中心并赋予了类标签后，按照密度从大到小的顺序，依据规则“样本的标签与密度大于它且离其最近的样本一致”，依次为剩余无标签样本分配相应的标签。

为便于理解，通过表 2.3、图 2.1 和图 2.2 中的示例，对 DPC 选取类簇中心

的过程可视化。表 2.3 为二类别含噪声的人工数据集的信息，样本的分布情况如图 2.1 所示，类簇中心选取决策图如图 2.2 所示。在图 2.1 和图 2.2 中，圆形表示多数类样本，菱形表示少数类样本，矩形表示噪声样本。

表 2.3 数据信息

序号	ρ_i	δ_i	γ_i	序号	ρ_i	δ_i	γ_i
1	2.985	1.462	4.364	13	0.935	0.138	0.129
2	1.918	0.801	1.537	14	1.031	0.120	0.124
3	2.532	0.098	0.249	15	1.699	0.072	0.123
4	2.128	0.113	0.240	16	0.867	0.136	0.118
5	2.219	0.097	0.216	17	0.789	0.144	0.114
6	1.749	0.120	0.210	18	0.817	0.135	0.110
7	1.720	0.107	0.184	19	0.673	0.156	0.105
8	1.497	0.116	0.174	20	0.809	0.128	0.103
9	1.220	0.141	0.172	21	0.449	0.147	0.066
10	1.716	0.081	0.139	22	0.156	0.196	0.031
11	1.057	0.129	0.137	23	0.001	0.447	0.001
12	1.026	0.131	0.135	24	0.000	0.601	0.000

由表 2.3 可知，样本 1 和样本 2 的γ值要显著大于其他样本点，而噪声样本 23 和 24 的γ值显著小于其他样本。

从图 2.1 中可以看出，γ值较大的样本 1 和样本 2 分别位于两个类簇的中心位置，而γ值较小的噪声样本 23 和 24 则远离类簇中心。

从图 2.2 可以看出，大多数样本都位于下方，噪声样本因为ρ值较小和δ值较大而位于左下方，样本 1 和样本 2 的ρ和δ值都比较大，因此它们位于离其他样本较远的右上角。若要将类簇划分为两类，则可以选取样本 1 和样本 2 作为这两个类簇的中心。

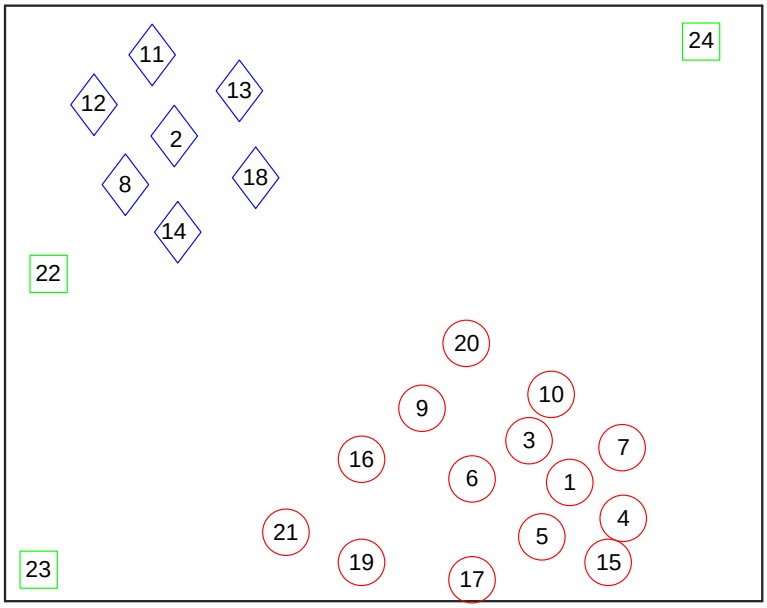


图 2.1 样本分布

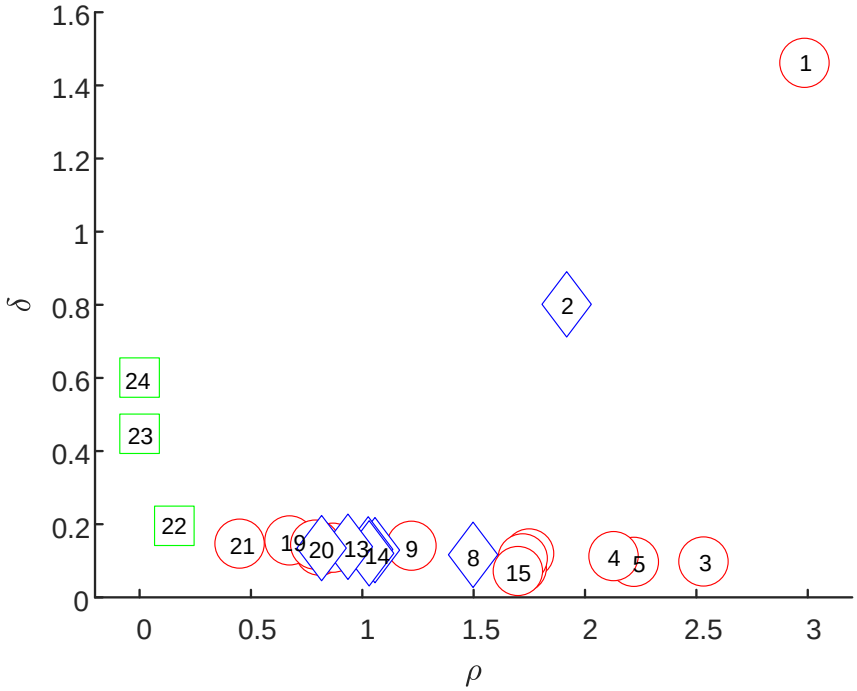


图 2.2 决策图

DPC 依靠人工选取类簇中心，显然这样的选取方式不够精准而且不能自动化地处理数据，针对以上问题，大量研究者对其进行了改进[22,76-78]。

DPC 的特性在于类簇中心选取不依赖先验知识，并且能利用密度峰值隐含

的层级关系来快速地分配类标签。δ被定义为“低密度点与离它最近的更高密度点之间的距离”，因此δ值的计算隐含了低密度点与高密度点之间的层级关系，在已知更高密度样本点的类标签情况下，可以通过递归的方式，快速地对其相关联的低密度样本点分配标签。因此，DPC 可以不受类簇形状约束，能够在非球形数据上进行聚类。

2.2.3 球 K 均值聚类算法

为减少 K-Means 迭代过程中的质心距离计算量，Xia S 等人[26]提出了球 K 均值聚类算法(Ball K-Means)。Ball K-Means 用“球簇”来表示类簇，并将球簇划分成“稳定区域(Stable Area)”和“活动区域(Active Area)”，将“样本点与所有质心的距离计算过程”用“样本点与其近邻球簇质心的距离计算过程”替换，该过程极大地减少了样本点与质心的距离计算量。

令O_k和O_l表示两个不同的球簇，o_k和o_l表示这两个球簇的质心，r_k和r_l表示这两个球簇的半径。o_k和r_k的计算如下：

$$o_k = \frac{1}{|O_k|}\sum_{x_i \in O_k} x_i \tag{2-11}$$

$$r_k = \max(\|x_i - o_k\|) \tag{2-12}$$

若$r_k > 0.5\|o_k - o_l\|$，则O_l为O_k的近邻球簇。显然，球簇之间的“近邻关系”是非对称的，即O_l可能是O_k的近邻球簇，但O_k可能不是O_l的近邻球簇。

令NB_{O_k}表示由O_k的全部近邻球簇的质心构成的集合，SA_k表示O_k的稳定区域，则O_k的稳定区域SA_k是以o_k为中心，以SAr_k为半径构成的球形区域，SAr_k的定义如下：

$$SAr_k = 0.5\min(\|o_k - o_l\|),\ o_l \in NB_{O_k} \tag{2-13}$$

球簇内“稳定区域”以外的区域为“活动区域”，若球簇的近邻球簇多于一个，“活动区域”会被细分为多个逐层嵌套的“环形区域”。

假设O_l不是O_k的近邻球簇，则$r_k \le 0.5\|o_k - o_l\|$，对O_k内任意的样本x_i有：

$$\begin{aligned}&\|o_k - x_i\| \le r_k \le 0.5\|o_k - o_l\| = 0.5\|o_k - x_i + x_i - o_l\| \\ &\le 0.5\left(\|o_k - x_i\| + \|x_i - o_l\|\right) \\ &\Rightarrow \|o_k - x_i\| \le 0.5\left(\|o_k - x_i\| + \|x_i - o_l\|\right) \\ &\Rightarrow \|o_k - x_i\| \le \|o_l - x_i\|\end{aligned} \tag{2-14}$$

假设 O_l 为 O_k 的近邻球簇，则 $r_k > 0.5\|o_k - o_l\|$，且存在 O_k 内的样本 x_i 使得 $\|o_k - x_i\| > 0.5\|o_k - o_l\|$。设 x_i 是线段 $o_k o_l$ 上一点，且满足条件 $\|o_k - x_i\| > 0.5\|o_k - o_l\|$ 和 $\|o_k - x_i\| + \|o_l - x_i\| = \|o_k - o_l\|$，则有：

$$\begin{aligned}&\Rightarrow 0.5(\|o_k - x_i\| + \|o_l - x_i\|) < \|o_k - x_i\| \\ &\Rightarrow \|o_l - x_i\| < \|o_k - x_i\|\end{aligned} \tag{2-15}$$

由式(2-14)和(2-15)知：在当前轮次迭代过程中，球簇“活动区域”内的样本可能会被划分到近邻球簇中，而球簇内所有样本均不会被划分到非近邻簇中。因此在每轮迭代过程中，只需计算球簇“活动区域”内的点与近邻球簇质心之间的距离，从而减少了距离的计算量，提高了算法的运行效率。

2.3 半监督学习介绍

半监督学习(Semi-Supervised Learning)是一种介于监督学习和无监督学习之间的机器学习方法。它可利用少量有标签样本中的先验信息指导无监督学习，也可利用大量无标签样本的潜在结构信息协助监督学习，减弱有标签样本不足对学习性能的影响，在不依赖外部资源的情况下最大化模型的学习效果。狭义上的半监督学习是指归纳半监督学习（纯半监督学习），广义上还包括直推学习。区别于直推学习方法只能预测输入数据中无标签样本的类别，纯半监督学习还能输出模型，可以对新样本进行预测。

本节主要介绍半监督学习的理论基础、学习任务、经典算法和两种与本文研究相关的自训练算法。

2.3.1 半监督学习理论基础

半监督学习是为解决实际问题中有标签样本不足而存在，在存在大量无标签样本的情况下，要发挥它们的作用需要使模型满足一些假设，如果假设不合理，

不仅不能提升学习性能还可能降低学习性能。

半监督学习的常用假设包括平滑假设、聚类假设和流形假设。平滑假设是指位于稠密区域内的两个邻近样本属于同一类别的概率大，而被稀疏区域分隔的两个样本属于同一类别的概率较小。聚类假设是指属于同一个簇或者是相互接近的两个样本属于同一个类别的概率更大，聚类假设可以看作是平滑假设的特例，它侧重于样本空间的全局特征，并利用大量无标签样本找到样本空间重点稀疏和稠密区域，从而优化决策边界[79]。流形假设是指在高维数据嵌入的低维流形的局部邻域内，两个样本属于同一类别的概率更大。流形假设利用大量无标签样本增加样本空间的密度，从而更准确地获取样本空间的局部特征[80]。

2.3.2 半监督学习任务类型

半监督学习算法的学习任务包括半监督分类、半监督聚类、半监督降维和半监督回归。半监督分类(Semi-Supervised Classification)是在充分发挥无标签样本潜在信息的作用下，获得比单独利用有标签样本所训练的分类器更强的分类器。相较于仅利用无标签样本的潜在信息进行类簇划分，半监督聚类(Semi-Supervised Clustering)在有标签样本的监督信息的帮助下能获得更好的聚类效果。半监督降维(Semi-Supervised Dimensionality Reduction)是在有标签样本的监督信息帮助下，找到高维输入数据的低维结构，同时保持原始高维数据结构不变。半监督回归(Semi-Supervised Regression)是在无输出的输入的帮助下，训练有输出的输入，相较于仅利用有输出的输入训练得到的回归器，它能获得性能更好的回归器。

2.3.3 半监督分类经典算法

半监督分类是半监督学习中最常见的一种任务，其代表方法有自训练方法、基于分歧的方法、生成式方法、判别式方法和基于图的方法。

（1）自训练方法

自训练是一种开放性、灵活性的算法框架，它对内容多样的半监督学习任务具有非常强的适应能力。自训练方法通常利用基分类器生成无标签样本的“伪标

签”，置信度较高的无标签样本及其“伪标签”将被用于扩展训练集，然后基于扩展训练集训练基分类器。自训练方法的基本步骤见表 2.4。

表 2.4 自训练方法的基本步骤

输入：	有标签样本集 $L=\{(x_1,y_1),(x_2,y_2),\cdots,(x_l,y_l)\}$，无标签样本集 $U=\{x_1,x_2,\cdots,x_u)\}$，基分类器 H
输出：	基分类器 H
步骤 1：	在 L 中训练分类器 H；
步骤 2：	从 U 中选取高置信度样本构成集合 S，由 H 赋予 S 内样本“伪标签”；
步骤 3：	将 S 添加进 L 中，并从 U 中移除；
步骤 4：	若 H 稳定或 $U=\varnothing$ 输出 H，否则转至步骤 1。

自训练方法的分类性能取决于高置信度样本的选取质量和基分类器的性能，如果高置信度样本选取不当，“伪标签”变成“错误标签”则会污染训练集，从而使得基分类器性能下降。

（2）基于分歧的方法

基于分歧的方法(Disagreement-Base methods)使用多个分类器实现无标签样本的充分利用，在学习过程中，将无标签样本作为多个分类器间信息交互的平台，从而达到协同训练的目的[81]。基于分歧的方法又可以分为多视图学习和单视图学习。

多视图学习基于以下假设[82]：在样本空间中，样本具有两个充分冗余且满足条件独立性的视图，视图表示对于一个样本的两种不同的描述，其本质上就是样本的属性集，不同视图可以给出关于样本的不同类型的区分信息。多视图学习的经典方法是协同训练(Co-Training)算法，协同训练的思想是利用多个拟合良好的分类器之间的差异性提高泛化能力，其基本步骤是：首先基于有标签样本在每个视图上都训练一个初始分类器，然后让每个分类器去挑选分类置信度最高的样本并赋予伪标签，将带有伪标签的样本传给另一个分类器去学习，从而在互相协助中不断增强分类器性能。

在实际应用场景中，通常多视图学习的假设难以满足，而单视图的数据却很

常见，因此，单视图学习逐渐受到研究者重视。单视图学习使用不同的分类器，或者不同的训练子集，或者使用不同的参数设置来生成不同的分类器。它无需多视图数据，仅仅要求基分类器之间具有显著的分歧（差异），即可通过相互提供伪标签样本的方式提升分类器泛化性能。不同视图、不同分类器、不同训练集、不同参数设置等都是产生差异的方式，但都不是必备条件。

（3）生成式方法

生成式方法是直接基于生成式模型的方法，它假设所有样本都是由同一潜在模型“生成”。这个假设通过潜在模型的参数将无标签样本与学习目标联系起来[83]，无标签样本隶属于每个类别的概率视作一组缺失的参数，然后通过一些迭代算法求解模型参数的最大似然估计[79]。

不同生成式方法的区别在于使用了不同的生成式模型作为基分类器，例如混合高斯模型（Mixture of Gaussians）、混合专家模型（Mixture of Expert）和朴素贝叶斯（Nave Bayes）等。基于生成模式的半监督学习方法简单、直观，但是当模型假设与数据分布不一致时，使用大量的无标签样本来估计模型参数反而会降低学得模型的泛化能力，寻找合适的生成式模型来为数据建模需要大量领域知识，这使得基于生成式模型的半监督学习在实际场景中应用有限[84]。

（4）判别式方法

判别式方法在有标签样本和无标签样本中学习决策边界，使其通过低密度数据区域，且使学习得到的分类超平面到最近的样本的距离间隔最大[79]。半监督判别式模型的代表性方法是半监督支持向量机（Semi-Supervises Supported Vector Machine, S3VM），它是支持向量机在半监督问题上的推广，旨在寻找能够将不同类别样本分开且穿过低密度样本区域的超平面。

半监督支持向量机最著名的一种算法是转导支持向量机(Transductive Support Vector Machine, TSVM)。TSVM 采用局部搜索的策略来进行迭代求解，即首先基于有标签样本训练出一个初始 SVM，接着使用该分类器对无标签样本进行预测，迭代交换决策边界两侧样本的标签以使间隔最大化，基于此更新分类器，从而实现在尽量正确分类有标签样本的同时，将决策边界移至样本分布相对更稀疏的区域[84]。

（5）基于图的方法

基于流形假设的半监督学习算法要求决策边界在数据嵌入到的低维流形上平稳地变化，由于实际训练样本的流形结构通常是未知的，研究者使用定义在训练样本的数据图去刻画数据的低维流形结构，由此提出基于图的半监督学习算法[80]。

基于图的方法的实质是标签传播，它基于流形假设，根据样本之间的几何结构构造边，用图的结点表示样本，利用图上的邻接关系将类标签从有标签样本向无标签样本传播[79]。基于图的方法通计算复杂度较高，对异常图结构缺乏鲁棒性，主要方法有最小分割方法、标签传播算法（LPA）和流形方法(Manifold method)等。根据标签传播方式的区别可将基于图的半监督学习方法分为两大类：一类方法通过定义满足某种性质的标签传播方式来实现显式标签传播，例如基于高斯随机场与谐函数的标签传播、基于全局和局部一致性的标签传播等；另一类方法则是通过定义在图上的正则化项实现隐式标签传播，例如通过定义流形正则化项，强制预测函数对图中的近邻给出相似输出，从而将标签从有标签样本隐式地传播至无标签样本[83,85]。

2.3.4 半监督模糊 C 均值优化的自训练算法

Gan 等人提出 STSFCM[12]算法，在 Self-Training 中集成半监督模糊 C 均值聚类(SSFCM[13,14])，利用聚类方法揭示无标签样本中的潜在空间结构，以协助训练一个更优的分类器。

STSFCM 将 SSFCM 和 SVM 组合，在每一轮迭代中：先由 SSFCM 计算无标签样本 $x_j \in U$ 的隶属度 b_{ij}，其中 $i \in [1,k]$，k 为类别数量，ε_1 为隶属度阈值，从 U 中选取 $b \geq \varepsilon_1$ 的样本构成样本集 S'；然后在有标签样本集 L 上训练一个 SVM 分类器 H，输出 S' 内样本的置信度 $f(x)$，$f(x) \geq \varepsilon_2$ 的样本构成高置信度样本集合 S''，其中 ε_2 为置信度阈值。若 $S'' = \varnothing$ 则在下一迭代过程中减小 ε_1 的值。

2.3.5 结合密度峰值和切边权值的自训练算法

卫丹妮等人在 STDP[25]算法的基础上结合 CEWS 方法，使用假设检验选取

高置信度样本，提出一种结合密度峰值和切边权值的自训练算法(STDPCEW[27,28])。

首先，STDPCEW 利用 DPC 发现样本的潜在层次结构，找到有标签样本的“前驱”样本集S_1和“后继”样本集S_2。其次，在$L \cup S_1$中构造相关邻近图，图中顶点满足$\forall k \neq i, j$使$d_{ij} \leq \max(d_{ik}, d_{jk})$，计算$S_1$内样本点的割边权重和$J_i$：

$$J_i = \sum_{j=1}^{n_i} W_{ij} I_{ij} \tag{2-16}$$

其中，W_{ij}为割边权重，I_{ij}的值为 0 或 1，当x_i的标签Y_i等于x_j的标签Y_j时$I_{ij} = 0$，否则$I_{ij} = 1$。J_i在零假设H_0下服从正态分布$N(\mu, \sigma^2)$，μ和σ^2的计算如下：

$$\mu = (1 - p_{Y_i}) \sum_{j=1}^{n_i} W_{ij} \tag{2-17}$$

$$\sigma^2 = p_{Y_i}(1 - p_{Y_i}) \sum_{j=1}^{n_i} W_{ij}^2 \tag{2-18}$$

其中p_{Y_i}表示样本标签为$\boldsymbol{Y}_i$的概率。

STDPCEW 通过设定一个显著性水平θ计算出左拒绝域，如果J_i的值落在左拒绝域内，则表明该样本预测正确，否则预测错误。然后找出所有落在左拒绝域内的样本构成高置信度样本集$\boldsymbol{S}_1'$，重新训练分类器。最后，使用相同的方法，从$\boldsymbol{S}_2$中选取高置信度样本集$\boldsymbol{S}_2'$，最终输出优化后的分类器。

2.4 本章小结

本章对分类学习、聚类学习和半监督学习等相关理论进行了简要介绍。首先介绍了四种类型的分类问题、四种经典分类算法、一种不平衡数据分类算法 USCBoost 以及多个分类性能评价指标。然后介绍了聚类算法的五种基本类型、密度峰值聚类算法和球 K 均值聚类算法。最后对半监督学习的理论基础、四种半监督学习任务、五种经典的半监督分类方法以及两种自训练方法 STSFCM 和 STDPCEW 进行了介绍。

3 密度峰值优化的自训练算法

本章利用密度峰值优化高置信度样本选取过程，提出了密度峰值隶属度优化的自训练算法(STDPM)。STDPM 利用密度峰值信息构建原型树结构，基于此结构可以快速搜索出潜在的高置信度样本；利用无标签样本与有标签样本的密度峰值信息定义了密度峰值隶属度，通过隶属度阈值从潜在的高置信度样本中筛选出高置信度样本。基于 8 个数据集上的对比实验，结果验证了 STDPM 的有效性。

3.1 问题描述

Self-Training 迭代过程中，较高置信度的无标签样本将被用于扩展训练集。然而，如果带有错误标签的样本被添加进训练集，误分风险会在迭代过程中不断累积，导致最终的分类器性能降低。

Li 等人[7]利用假设检验选取高置信度样本，提出了编辑自训练算法(SETRED)。SETRED 采用割边权重统计方法，将含有较多割边的样本从训练集中移除，因此它的计算开销较大，而且当样本在相关邻近图(RNG)上的近邻数量较少或者割边权重不平衡时，假设检验的效果不佳。聚类算法[10,11]可以从大量无标签样本中提取有价值的信息来辅助高置信度样本的选取。Gan 等人[12]基于聚类假设，提出了基于半监督模糊 C 均值聚类的自训练算法(STSFCM)。STSFCM 将 SSFCM 输出的类簇隶属度中大于设定阈值的样本作为高置信度样本。STSFCM 算法从训练集中移除类簇归属确定性低的样本是一种有效的思路，然而它对非球形数据的适应性不好，而且易受初始点选择的影响，因此，有研究者尝试在 Self-Training 中嵌入 DPC 算法。DPC 对非球形数据具有较好的适应性，可以将任意维度数据映射成二维，在二维空间内建构数据之间的层次关系[16]。聚类假设主要考虑数据的整体特性，而流形假设主要考虑数据的局部特性。为兼顾数据的整体和局部特性，Wu 等人[25]提出了基于数据密度峰值的自训练半监督分类算法(STDP)。STDP 根据密度峰值揭示的空间结构，先将有标签样本的前驱作为高置信度样本，再将有标签样本的后继作为高置信度样本，这种高置信度样本选取方法速度极快。卫丹妮等人[27]提出了 STDPCEW 算法，在 STDP 的基础上结合 CEWS 方法，使用数据编辑技术对选取的高置信度样本进一步优化。STDPCEW 和

SETRED 均需要构造 RNG，导致算法整体复杂度较高，难以在大规模数据集中应用。因此，针对上述方法存在的一些问题，本章提出了一种可以适用于非球形数据、具有较高运行效率和分类性能的算法——密度峰值隶属度优化的自训练算法。

3.2 密度峰值隶属度优化的自训练算法

3.2.1 原型树和近亲结点

由式(2-9)知，DPC 中的“峰值”是低密度点与距其最近的更高密度点之间的距离，峰值融合了密度、距离和引导信息。假设存在一个样本 x_j，已知所有样本的局部密度和距离，则能够通过遍历样本，找到距 x_j 最近的一个低密度样本点 x_i，并称该更高密度的样本点 x_j 为 x_i 的原型，形式化定义如 3.1 所示。

定义 3.1 样本 x_i 的原型 P_i 为：

$$\begin{aligned} &P_i=x_j \\ &\text{s.t. } \exists j, \delta_i = \min_{j:\rho_j>\rho_i} d_{ij} \end{aligned} \tag{3-1}$$

对于密度最高的样本点，由于不存在比它密度更高的点，其原型为自身，即 $P_i=x_i, \forall j \neq i, \rho_i > \rho_j$。

根据样本点与其原型之间的层次关系递归构造出原型树，主要步骤如下：

（1）确定原型树的根结点：遍历样本集，搜索出密度最高的样本点，将其作为根结点，根结点 x_r 须满足条件 $\forall j \neq r, \rho_r \geq \rho_j$。

（2）确定根结点之外的结点：遍历样本集，按照局部密度从高到低的顺序，根据原型找到与原型关联的子结点，通过递归的方式自顶向下确定所有结点。

为便于理解，通过表 3.1、图 3.1 和图 3.2 的示例可视化原型树结构。构造包含 3 类别的高斯分布样本，每个类别包含 5 个样本，样本密度峰值等信息如表 3.1 所示，为便于比较，ρ 和 δ 值已经归一化。

表 3.1 样本信息

标记	ρ_i	δ_i	γ_i	P_i
1	1.000	1.000	1.000	1
2	0.690	0.747	0.515	12
3	0.513	0.571	0.293	2
4	0.443	0.183	0.081	15
5	0.831	0.028	0.023	1
6	0.301	0.068	0.020	7
7	0.496	0.034	0.017	3
8	0.223	0.059	0.013	5
9	0.160	0.077	0.012	2
10	0.443	0.022	0.010	4
11	0.055	0.110	0.006	3
12	0.829	0.001	0.001	1
13	0.002	0.155	0.000	11
14	0.000	0.186	0.000	12
15	0.639	0.000	0.000	2

图 3.1 为样本分布和原型树，图中圆形、矩形和六边形表示不同类别的样本，箭头指向样本点的原型。虚线圆内区域为类簇局部中心，由表 3.1 知，样本 1、样本 2 和样本 3 的 ρ 和 δ 值较大，γ 值也较大，所以它们可以成为各自所在类簇的中心点。

由表 3.1 知，样本 1 的 ρ 值最大，所以其作为根结点，样本 5 和样本 12 的原型为样本 1，因此，样本 5 和样本 12 作为样本 1 的子结点。通过遍历所有样本，根据每个样本的原型，可以递归构造图 3.2 所示的原型树。

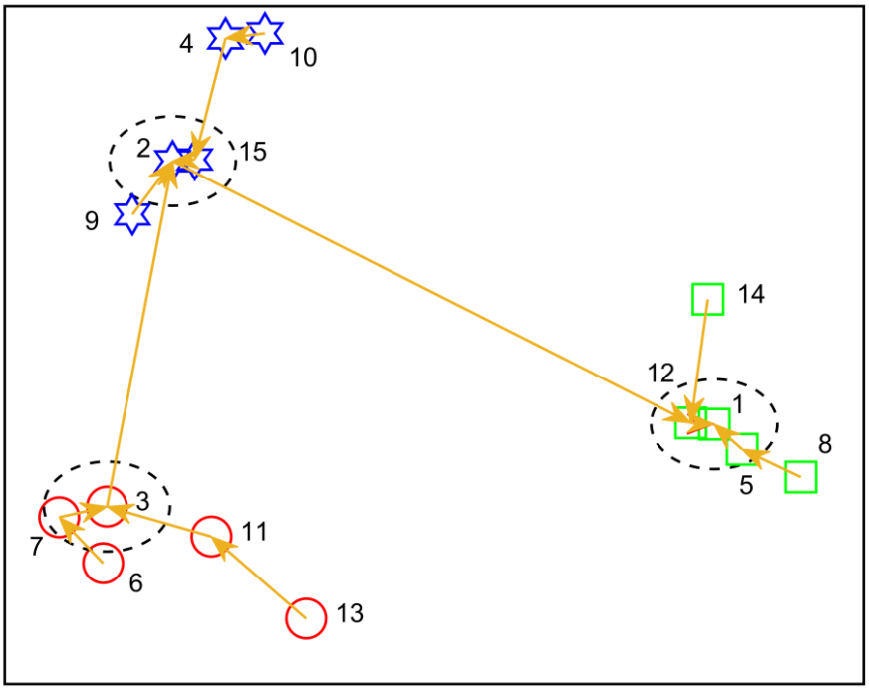


图 3.1 三类别样本的分布和原型树

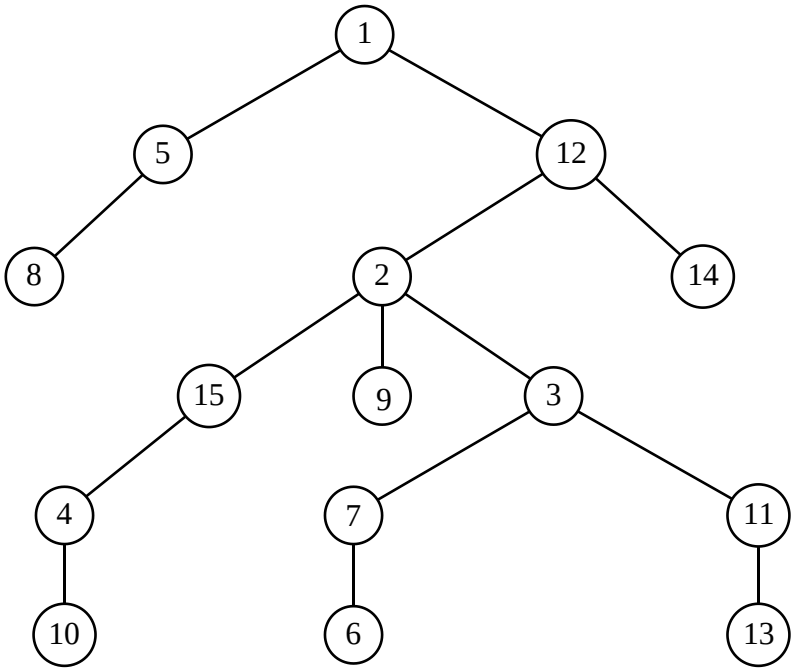


图 3.2 原型树示意图

由图 3.1 和图 3.2 可知，原型树中各结点与其原型之间形成了树状的层次结构，各结点之间存在父子结点和兄弟结点关系，这些关系可以统称为“近亲关系”，

这些结点称为“近亲结点”，如下给出近亲结点的定义。

定义 3.2 样本 x_i 的近亲结点为：

$$r_i = \{x_k \mid x_k = P_i \vee x_i = P_k \vee P_i = P_k, i \neq k\} \tag{3-2}$$

样本 x_i 的全部近亲结点构成的集合为“近亲结点集”，其定义如下。

定义 3.3 样本 x_i 的近亲结点集为：

$$R_i = \{r_1, r_2, \cdots, r_m\} \tag{3-3}$$

式(3-3)中，m 是 x_i 的近亲结点数量。如图 3.2 所示，样本 3 的父结点为样本 2，兄弟结点为样本 9 和 15，子结点为样本 7 和 11，则样本 3 的近亲结点集 $R_3 = \{x_2, x_7, x_9, x_{11}, x_{15}\}$。

在原型树中搜索每个有标签样本的近亲结点，获得近亲结点集 $R = R_1 \cup R_2 \cup \cdots \cup R_l$，其中 l 为有标签样本的个数，R 中既包括无标签样本也包括有标签样本。从 R 中移除有标签样本，无标签近亲结点集 $R_U = R - L$。

3.2.2 密度峰值隶属度

在 K-Means 等硬聚类算法中，样本属于某个类是确切和唯一的，它要么属于某个类要么不属于某个类，样本的类簇隶属度为 0 或 1。而在 FCM 等软聚类算法中，样本的隶属度是模糊的，其取值范围为 $[0,1]$。假设将数据 X 聚成 m 个类别，对任意的 j 和 k，$j \neq k$，$j,k \in [1,m]$，如果样本 x_i 的隶属度 $U_{ij} \gg U_{ik}$，则 x_i 是第 j 类的概率最大，并且 x_i 被分类器误分的可能性也会更小。反之，如果样本对不同类的隶属度相差较小，则其被误分的可能性更大。因此，STDPM 通过设定一个阈值，将隶属度大于阈值的样本作为高置信度样本。

STSFCM 利用 SSFCM 来计算样本的类簇隶属度，但是 SSFCM 依赖质心距离计算，在非球形数据上的性能会受影响。本文充分考虑了数据的局部结构，提出了一种密度峰值隶属度，可以适用于非球形数据。

给定无标签样本点 x_i 和有标签样本点 x_k，x_i 的簇峰值定义如下：

定义 3.4 样本 x_i 的簇峰值为：

$$\zeta_{ij} = \min_{k:\rho_k \geq \rho_i, y_k = M_j} d_{ik}, \quad j = 1,2,\cdots,m \tag{3-4}$$

其中 m 表示类簇的个数，y_k 表示样本 x_k 的标签，M_j 表示第 j 类的标签，d_{ik} 表示 x_i 与 x_k 的距离。

图 3.3 所示为簇峰值示意图，图中不同图形表示不同的类簇 M_j，有填充的图形表示有标签样本，无填充的图形表示无标签样本。如图 3.3 所示，在簇 M_1、M_2 和 M_3 中，距样本 11 最近且密度更高的有标签样本分别为样本 2、3 和 12，则样本 11 对应不同类簇的簇峰值分别为 $\zeta_{11,1} = d_{11,2}$，$\zeta_{11,2} = d_{11,3}$，$\zeta_{11,3} = d_{11,12}$。

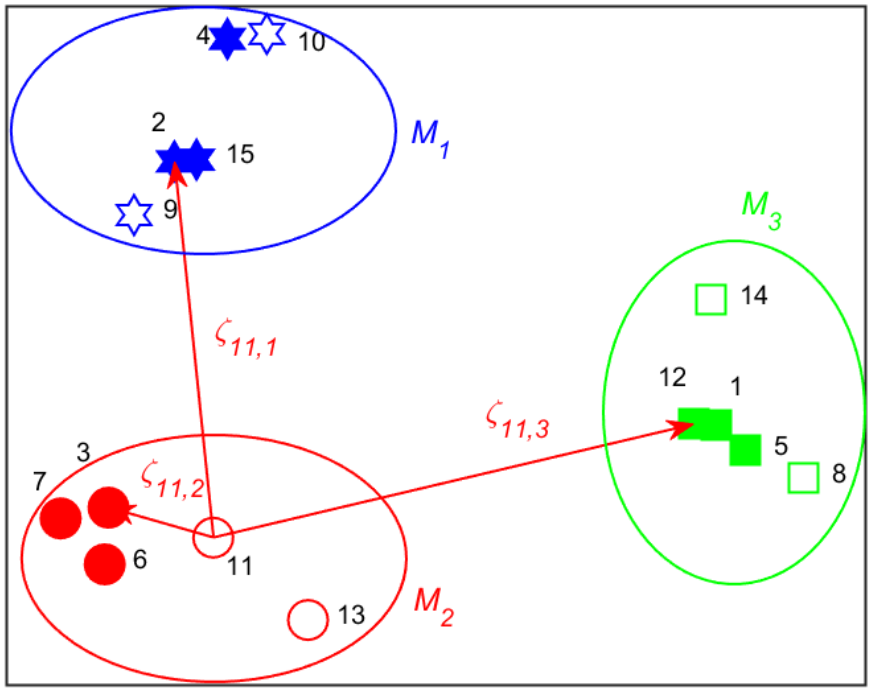


图 3.3 簇峰值示意图

根据簇峰值计算类簇隶属权重，给定样本 x_i 隶属第 j 类的权重为：

$$W_{ij} = \zeta_{\Delta} - \zeta_{ij} \tag{3-5}$$

其中 ζ_{Δ} 是 ζ 中的最大值。由式(3-5)知，ζ_{ij} 越大 W_{ij} 就越小，样本 x_i 属于第 j

类的概率就越小。将类簇隶属权重W按行归一化后得到密度峰值隶属度。密度峰值隶属度的定义如下：

定义 3.5 样本x_i的密度峰值隶属度为：

$$\xi_{ij}=\frac{\zeta_{ij}-\zeta_{\vee}}{\zeta_{\wedge}-\zeta_{\vee}} \tag{3-6}$$

其中$\zeta_{\wedge}$为$\zeta_{i\cdot}$中的最大值，$\zeta_{\vee}$为$\zeta_{i\cdot}$的中最小值。

样本的密度峰值隶属度依赖于它与不同类簇中有标签样本的距离，充分利用了样本的密度峰值信息，能够适用各种形状的数据。

STDPM 将无标签近亲结点集中密度峰值隶属度值较高的样本作为高置信度样本，并将其用于迭代训练分类器。STDPM 的算法步骤如表 3.2 所示。

表 3.2 STDPM 算法描述

输入：	有标签样本集L，无标签样本集U，截断距离比例阈值α，隶属度阈值β
输出：	分类器H。
步骤 1	由式(3-1)计算原型P，构造原型树；

步骤 2	While $U\neq\varnothing$ Do:
	基于L训练基分类器H，初始化高置信度样本集$S=\varnothing$；
	搜索L内样本的无标签近亲结点集R_U；
	由式(3-6)计算密度峰值隶属度ξ；
	For $x_i\in R_u$ Do:
	If $\exists\xi_{ij}\geq\beta$:
	增加高置信度样本：$S\leftarrow S\bigcup x_i$；
	End If
	End For
	由H对S内的样本赋予伪标签，更新：$L\leftarrow L\bigcup S$，$U\leftarrow U-S$；
	End While

步骤 3	输出H

3.3 算法复杂度分析

令 n 为输入数据的样本数量，t 表示迭代次数。STDPM 计算密度峰值的时间复杂度为 $O(n^2)$，构造原型树的时间复杂度为 $O(n^2)$，计算密度峰值隶属度的时间复杂度为 $O(n^2)$，综上 STDPM 的整体时间复杂度为 $O(tn^2)$。SETRED 和 STDPCEW 的时间复杂度主要在于构造 RNG，其整体时间复杂度为 $O(tn^3)$。STDP 的时间复杂度主要在于计算密度峰值，其整体时间复杂度为 $O(tn^2)$，STSFCM 计算模糊隶属度的时间复杂度为 $O(n^2)$，训练 SVM 分类器的复杂度为 $O(n^3)$，其整体时间复杂度为 $O(tn^3)$。综上所述，相较于 SETRED、STSFCM 和 STDPCEW 算法，STDPM 的时间复杂度更低，但与 STDP 一致。

STDP、STDPCEW 和 STDPM 的空间复杂度主要在于计算密度峰值，整体空间复杂度均为 $O(n^2)$，SETRED 的空间复杂度主要在于构造相关邻近图，其整体空间复杂度为 $O(n^2)$，STSFCM 的空间复杂度主要在于计算模糊隶属度和 SVM 分类器的训练、预测，其整体空间复杂度为 $O(n^2)$。综上所述，STDPM 与其他 4 个算法的空间复杂度一致。

3.4 实验结果与分析

3.4.1 实验设置

本文设计了多个对比实验来验证 STDPM 的有效性，所有实验在以下环境中进行：酷睿 i5-10210U 处理器，8GB 运行内存，Windows10 x64 系统，MATLAB R2019a 应用程序。

实验选取 SETRED、STDP、STDPCEW 和 STSFCM 作为 STDPM 的对比算法，基分类器采用最近邻分类器(Nearest Neighbor, NN)，各算法的参数设置如表 3.3 所示。为保证结果可以比较，将 STSFCM 的基分类器 SVM 用 NN 替换，因此其阈值参数只有一个 ε。

表 3.3 参数设置

算法	参数
STDPM	α=2，β=0.5
SETRED[7]	θ=0.1
STDP[25]	α=2
STDPCEW[27]	α=2，θ=0.1
STSFCM[12]	ε=0.5

3.4.2 实验数据集

在选取的 8 个基准数据集上进行实验，数据集信息如表 3.4 所示。其中 Banknote、Breast、Hepatitis、Ionosphere、Palm、Segment 和 Zoo 数据集均来源于公开的 UCI 数据库①，Yale(32×32)数据集来源于公开的耶鲁大学人脸数据库②。

表 3.4 中大部分数据集的维度较高，为便于实验对比，先将维度大于 10 的数据集用主成分分析(Principal Component Analysis, PCA)降到 10 维，小于 10 维的数据集则保持不变。

表 3.4 数据集信息

数据集	样本数	属性数	类别数
Banknote	1372	4	2
Breast	699	10	2
Hepatitis	142	13	2
Ionosphere	351	34	2
Palm	2000	256	100
Segment	2310	19	7
Yale	165	1024	15
Zoo	101	16	7

① http://archive.ics.uci.edu/ml/index.php
② http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html

3.4.3 固定有标签样本比例的分类性能实验

在表 3.4 的数据集中测试 5 个算法的分类性能，实验结果如表 3.5 和表 3.6 所示。按以下方式进行 30 次实验：从原始数据集中随机抽取 10%作为有标签样本集，其余 90%作为无标签样本集，将它们作为算法的输入数据；各算法在每个数据集上运行，将原始数据集作为固定测试集，测试输出分类器的准确率(Accuracy)和 F1 分数(F1-Score)。

表 3.5 各算法的准确率(Accuracy)结果

数据集	算法				
	STDPM	SETRED	STDP	STDPCEW	STSFCM
Banknote	99.57	99.37	99.10	99.39	99.38
Breast	58.87	58.77	57.18	57.22	57.05
Hepatitis	61.33	59.23	58.50	53.77	56.65
Ionosphere	88.59	86.78	85.59	86.98	87.81
Palm	90.52	88.18	88.16	90.48	88.45
Segment	93.64	93.54	93.38	93.47	93.23
Yale	78.84	77.63	76.47	77.98	78.55
Zoo	91.29	87.73	87.93	88.48	87.60

由表 3.5 和表 3.6 可知，STDPM 在 Banknote、Breast、Hepatitis、Ionosphere、Palm、Segment、Yale 和 Zoo 数据集数据集上都获得了最高的分类性能，Accuracy 值分别高出最接近的算法 0.18%、0.1%、2.1%、0.78%、0.04%、0.1%、0.29%和 2.81%，F1-Score 值分别高出最接近的算法 0.14%、0.09%、2.44%、0.62%、0.13%、0.11%、1.83%和 5.33%。

因为大多数测试数据集分布都较平衡，所以 Accuracy 值和 F1-Score 值趋于一致。虽然 Zoo 数据集分布不平衡，但是 STDPM 的分类性能仍然高于对比算法。以上实验结果表明，STDPM 利用密度峰值定义密度峰值隶属度，通过密度峰值隶属度选取高置信度样本是有效的。

表 3.6 各算法的 F1 分数(F1-Score)结果

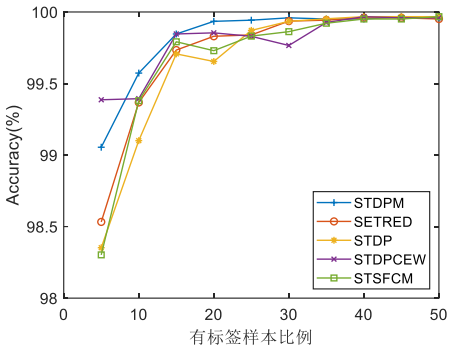
数据集	算法				
	STDPM	SETRED	STDP	STDPCEW	STSFCM
Banknote	99.35	99.14	98.96	99.19	99.21
Breast	58.51	58.42	57.43	57.51	57.21
Hepatitis	60.90	58.46	58.38	52.95	55.77
Ionosphere	87.50	85.97	84.58	86.24	86.88
Palm	88.11	84.54	84.59	87.98	85.03
Segment	93.33	93.22	93.05	93.14	92.89
Yale	62.88	58.92	55.79	57.30	61.05
Zoo	88.09	81.02	79.89	82.76	78.63

3.4.4 变动有标签样本比例的分类性能实验

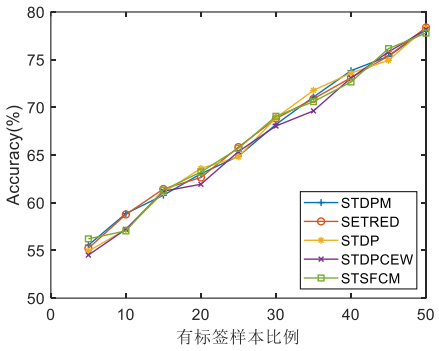
有标签样本占全部样本的比例过低会导致监督信息过少，但是增加有标签样本并不一定能增加有用信息，还可能增加有害信息。因此，分析有标签样本比例变动对各算法性能的影响是有必要的。

在表 3.4 的数据集上进行实验，初始有标签比例取值范围为 5%~50%，步长为 5%，各算法运行 30 次并记录 Accuracy 的均值结果。实验结果如图 3.4 所示。

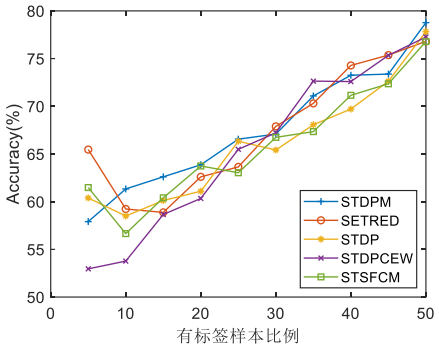
由图 3.4 可以看出，当有标签样本比例小于 20%时，STDPM 在 Banknote、Hepatitis、Ionosphere、Yale 数据集上能获得较高的准确率，这是因为 STDPM 能在有标签样本较少的情况下，利用密度峰值信息构造更接近真实数据结构的原型树，利用密度峰值隶属度选取出质量更高的样本添加进训练集。对于不同的初始有标签样本比例，在 Breast、Palm 和 Segment 数据集上各个算法的性能表现较为稳定，在 Hepatitis、Ionosphere、Yale 和 Zoo 数据集上，大部分算法的性能表现波动较大，而 STDPM 在 8 个数据集上的分类性能相较于 4 个对比算法更加平稳。



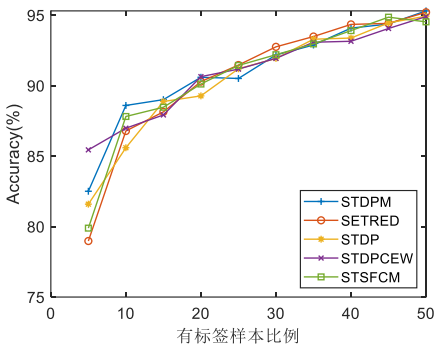
(a) Banknote



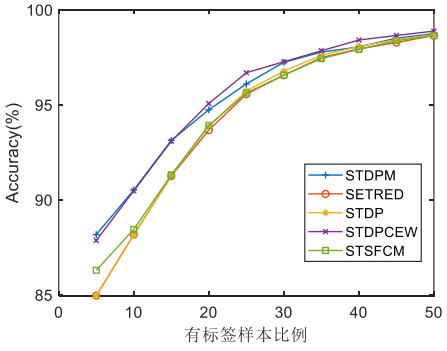
(b) Breast



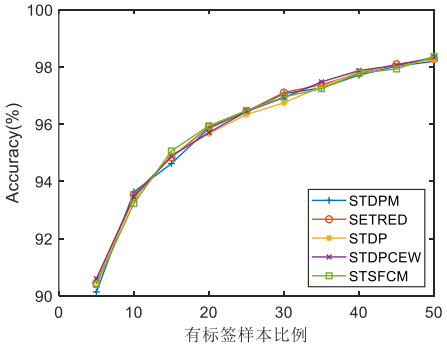
(c) Hepatitis



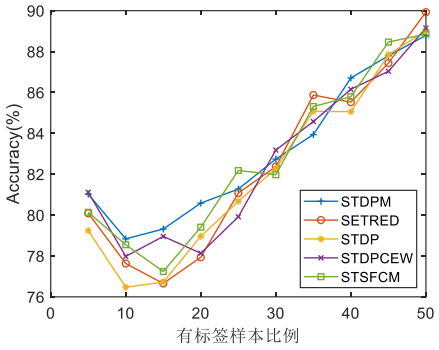
(d) Ionosphere



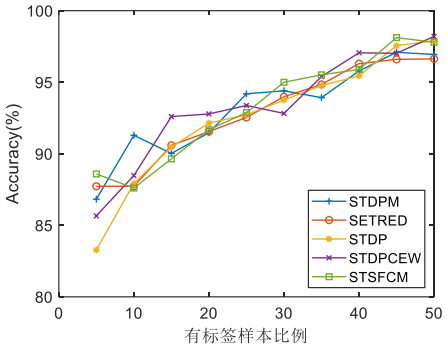
(f) Palm



(g) Segment



(h) Yale



(i) Zoo

图 3.4 不同有标签样本比例下各算法的准确率(Accuracy)

3.5 参数研究

STDPM 算法需要输入两个超参数：截断距离截取比例阈值α和密度峰值隶属度阈值β。α取值的变化会影响截断距离d_c的取值，使构造的原型树发生变化，最终对隶属度的计算产生影响。α取值过大会导致类簇合并，取值过小会导致局部中心点过多，因此，α取值不合适会导致构造的原型树偏离数据的真实结构。

β取值的变化会影响高置信度样本的选取，取值过大会导致迭代过程中选取高置信度样本过少，使迭代次数增加，取值过小会导致较低置信度的样本被添加进训练集，增加了误分风险。

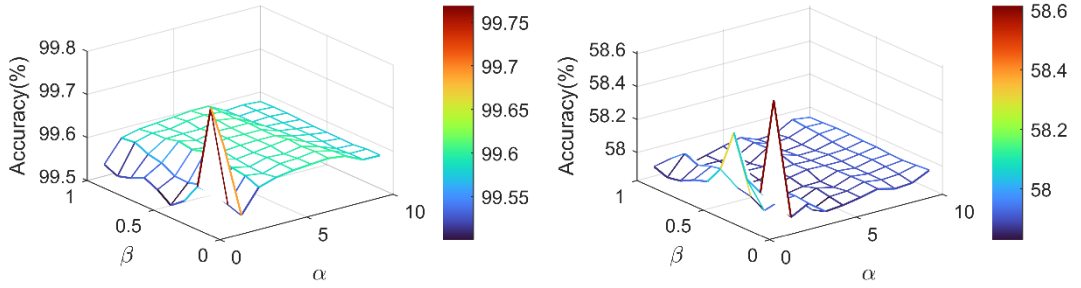
图 3.5 展示了超参数α和β的不同取值对 Accuracy 的影响，α的取值范围为$[1,10]$，步长为 1，β的取值范围为$[0.1,1]$，步长为 0.1。

如图 3.5 所示，在 8 个数据集上，α取值小于 5 能获得最高的准确率，在 Banknote、Breast、Hepatitis、Ionosphere 和 Yale 数据集上，α取值较大时准确率较低，这是因为截断距离过大导致类簇合并。因此，根据实验结果，α的建议取值范围为$(0.5,5]$。在 Banknote、Breast、Hepatitis、Yale 和 Zoo 数据集上，β取值小于 0.5 能获得较高的准确率，在 Ionosphere、Palm 和 Segment 数据集上，β取值大于 0.5 能获得较高的准确率。因此根据实验结果，β的合适取值需要根据具体的数据集来设置。

3.6 本章小结

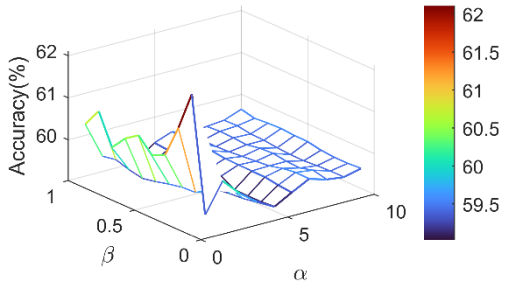
针对自训练算法中高置信度样本选取不当会导致分类性能下降的问题，本章提出了一种密度峰值隶属度优化的自训练算法(STDPM)。首先利用密度峰值信息构造原型树，在原型树中搜索有标签样本的无标签近亲结点集，利用近亲结点集内无标签样本的簇峰值信息计算密度峰值隶属度；通过设定的阈值选取密度峰值隶属度较大的无标签样本作为高置信度样本。基于 8 个基准数据集中的实验表明：STDPM 的分类性能优于 4 个对比算法，利用密度峰值隶属度能够选取更高

质量的高置信度样本，能够提升自训练算法的分类性能。

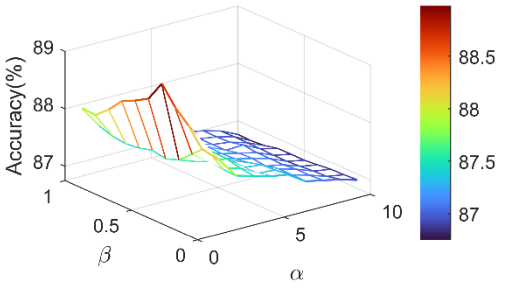


(a) Banknote

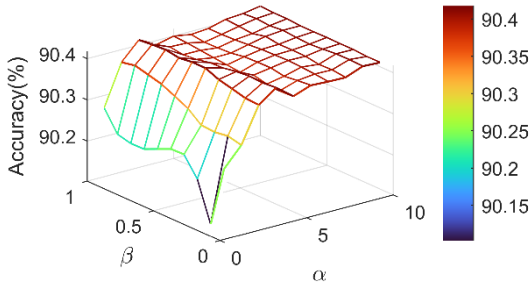
(b) Breast



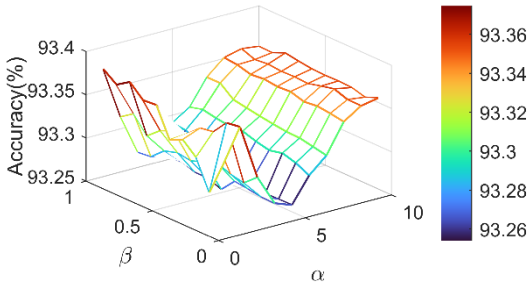
(c) Hepatitis



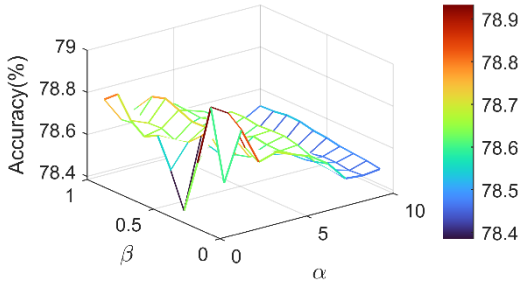
(d) Ionosphere



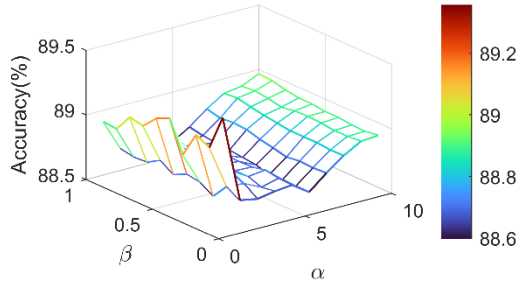
(f) Palm



(g) Segment



(h) Yale



(i) Zoo

图 3.5 不同参数 α 和 β 对准确率的影响

4 密度峰值优化的不平衡数据分类算法

本章利用密度峰值对欠采样方法和误分代价计算方法进行优化，提出了一种不平衡数据分类算法 DPBCPUSBoost。针对 RUS 方法会丢失决策边界区域内有价值样本的问题，提出了一种尽量保留了决策边界区域内有价值样本的欠采样方法(DPBCPUS)。DPBCPUS 根据密度峰值信息计算多数类样本的采样权重，并提高多数类球簇“易误分区域”内样本的采样权重，最后按照采样权重对多数类样本进行欠采样。针对 USCBoost 等算法没有考虑到不同样本之间应该具有不同误分代价的问题，提出了一种兼顾类依赖和样本依赖的误分代价计算方法：首先根据类别分布信息计算类依赖代价，然后根据密度峰值信息计算样本依赖代价，最后融合两种类型的代价作为样本的误分代价。在 10 个基准测试数据集上进行实验，结果验证了 DPBCPUSBoost 的有效性。

4.1 问题描述

传统分类算法对所有样本一视同仁，然而对于分布不平衡的数据，一些有价值的样本容易被忽视，因此需要采用一些策略来增加对更高价值样本的关注度。通常将欠采样方法与集成学习、代价敏感学习等方法结合，能够获得较高的分类性能和运行效率。

RUS 方法从多数类样本中随机移除一部分样本，以平衡少数类样本与多数类样本的数量。该方法简单粗暴、速度很快，但是它极有可能会删除一些有价值的样本，例如那些具有代表性的或者位于决策边界区域内的样本。针对上述问题，本章提出了一种密度峰值优化的球簇划分欠采样方法(DPBCPUS)，尽量保留了这些有价值的样本。

USCBoost 等代价敏感学习算法采用类依赖代价，这种代价计算方式依赖先验类别分布信息，只考虑了不同类之间的误分代价，没有区分不同样本的误分代价。针对以上问题，本章提出了一种考虑类依赖和样本依赖的误分代价计算方法，对更高价值的样本赋予了更高的误分代价。

4.2 密度峰值优化的球簇划分欠采样不平衡数据分类算法

4.2.1 密度峰值优化的球簇划分欠采样方法

DPC 算法对局部中心点的定义是：若一个样本点与其周围的点相比具有更高的密度，并且离其他高密度的点较远，则该样本点为局部中心点。因此，样本的γ值（局部密度和峰值的乘积）可以反映它成为局部中心的可能性，γ值越大，其作为局部中心的概率也越大。如表 2.3、图 2.1 和图 2.2 所示，γ值最大的两个样本点分别作为不同类簇的中心，而γ值较大的样本则成为局部中心，γ值较小的样本则分布在类簇边缘。

类簇的局部中心点一定程度上能够代表那些围绕它的样本点，具有揭示类簇结构的重要价值。因此，DPBCPUS 根据密度峰值信息定义样本的采样权重，尽量保留这些局部中心点，以降低欠采样造成的信息损失。

定义 4.1 样本x_i的采样权重为局部密度和峰值的乘积η_i：

$$\eta_i = \gamma_i = \rho_i \times \delta_i \tag{4-1}$$

通常决策边界区域内的样本易被误分，而远离决策边界的样本被误分的概率要小得多。因此，如果尽量保留决策边界区域内易被误分的样本，将使它们在分类器训练过程中获得更多关注，从而改善整体分类性能。

在 Ball K-means 方法中，球簇被划分为“稳定区域”和“活动区域”，“活动区域”内的样本被划分到近邻球簇的概率更大。受该方法的启发，DPBCPUS 将多数类球簇划分为“易误分区域”和“难误分区域”，少数类球簇则不进行划分。令$o_{\min}$表示少数类球簇的中心，o_{maj}表示多数类球簇的中心，r_{maj}为多数类球簇的半径，如下给出这两个区域的定义。

定义 4.2 多数类球簇的“易误分区域”是“以$o_{\min}$为中心，以r_{me}为半径的球形区域”和“以o_{maj}为中心，以r_{maj}为半径的球形区域”重叠的区域，标记为A_{me}，多数类球簇中除去重叠部分的区域为“难误分区域”，标记为A_{mh}。其中，半径r_{me}

定义如下：

$$r_{me} = 0.5\left\| o_{maj} - o_{min} \right\| \tag{4-2}$$

图 4.1 为划分球簇的示意图，实线表示球簇的边界，弧形虚线与多数类球簇边界线所围成的区域为“易误分区域”，多数类球簇内“易误分区域”之外的区域为“难误分区域”。

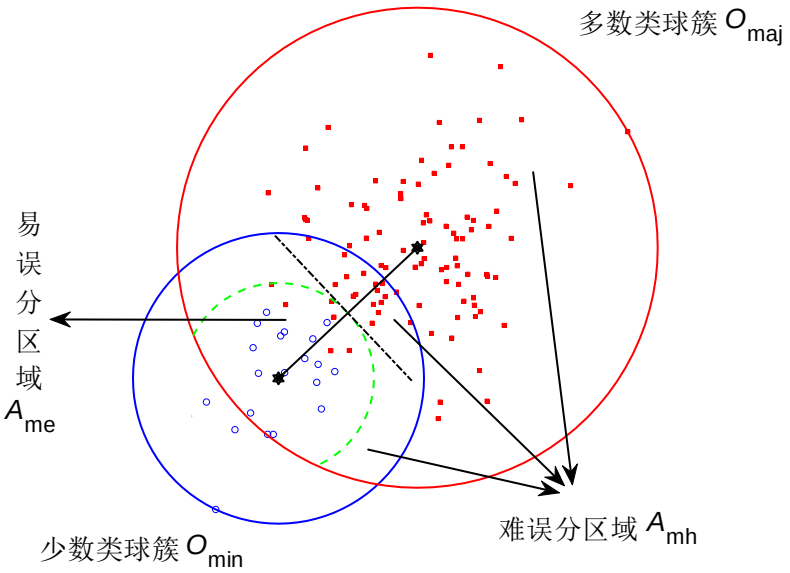


图 4.1 划分球簇

如果少数类球簇是多数类球簇的近邻球簇，则对多数类球簇中落入“易误分区域”内的样本的采样权重进行调整；如果少数类球簇不是多数类球簇的近邻球簇，或者样本落在“难误分区域”内，则不对采样权重进行调整。样本的采样权重计算如下：

$$\hat{\eta}_i = \begin{cases} \eta_i + 1, & x_i \in A_{me} \\ \eta_i, & x_i \in A_{mh} \vee o_{min} \notin NB_{O_{maj}} \end{cases} \tag{4-3}$$

其中：$\boldsymbol{x}_i \in O_{maj}$，$NB_{O_{maj}}$ 表示多数类球簇 O_{maj} 的近邻球簇的中心构成的集合。

将调整后的采样权重归一化，得到最终的采样权重：

$$\eta_i' = \hat{\eta}_t / \sum_{1}^{|O_{maj}|} \hat{\eta}_i \tag{4-4}$$

USCBoost 在初次迭代时采用 RUS 方法对多数类样本进行欠采样，当不平衡比例非常高时，初次迭代训练的分类器质量会非常差，进而影响之后的训练过程。

而 DPBCPUS 则按照权重η'对多数类进行欠采样，这种方式尽量保留了决策边界区域内易误分的多数类样本和更具有代表性的多数类样本[12,27]。

4.2.2 考虑类依赖和样本依赖的误分代价计算方法

在代价敏感学习的二分类问题中，样本的类依赖误分代价矩阵C可以由表 4.1 所描述。

表 4.1 二分类问题误分代价矩阵

预测类别 / 实际类别	正例	负例
正例	c_{11}	c_{10}
负例	c_{01}	c_{00}

表 4.1 中，c_{ij}表示将第i类样本预测为第 j 类的代价，$i, j \in \{0,1\}$。当$c_{ii} = c_{jj} = 0$时，即表示正确分类不会产生代价。通常在不平衡数据分类问题中$c_{10} > c_{01}$，即表示正例样本被预测为负例的代价要大于负例样本被预测为正例的代价。

USCBoost 采用类依赖误分代价，将多数类样本的误分代价定义为$|O_{\min}|/(|O_{\min}|+|O_{\mathrm{maj}}|)$，少数类样本的误分代价定义为$|O_{\mathrm{maj}}|/(|O_{\min}|+|O_{\mathrm{maj}}|)$，显然这种代价定义方式会使得少数类样本的误分代价高于多数类样本，但它没有考虑到同类样本内部之间的差异性。

由表 2.3、图 2.1 和图 2.2 可知，ρ和δ值越大的样本点越有可能成为局部中心点，而这些中心点在分类过程中的价值要更高，其被误分的代价也更大。因此，同时考虑类依赖代价和样本依赖代价，根据样本的类别分布信息和样本的类簇结构信息，对不同样本定义不同的误分代价。

定义 3 样本x_i的误分代价c_i为：

$$c_i = \begin{cases} \gamma_i + |O_{\min}|/n, & x_i \in O_{\mathrm{maj}} \\ \gamma_i + |O_{\mathrm{maj}}|/n, & x_i \in O_{\min} \end{cases} \tag{4-5}$$

本文受 DPC 和 Ball K-Means 思想启发，根据密度峰值信息计算采样权重，将多数类球簇划分为“易误分区域”和“难误分区域”，并增加“易误分区域”内样本的采样权重，在初次迭代时根据采样权重对多数类样本进行欠采样。本文融合密度峰值信息和样本类别分布信息，定义了新的误分代价，然后借鉴 AdaCost 和 USCBoost 中的代价调整方法，在迭代过程中进一步增加高误分代价样本的权重。DPBCPUSBoost 的算法步骤如表 4.2 所示。

表 4.2 DPBCPUSBoost 算法步骤

输入：	训练样本集 X，标签集 Y，迭代次数 T，弱分类器 h，截取比例阈值 α；
输出：	集成分类器 H；
初始化：	样本分布权重 $W_1=\{w_{11},w_{12},\cdots,w_{1N}\}$，$w_{1i}=1/n$，$i=1,2,\cdots,n$。
步骤 1	由式(4-4)和式(4-5)计算样本的采样权重 η' 和误分代价矩阵 C；

步骤 2	For $t\in[1,T]$ Do:
	If $t=1$:
	按采样权重 η' 对多数类进行欠采样；
	Else:
	选取多数类样本中权重较大的前 $\lvert O_{\min}\rvert$ 个样本；
	End If
	欠采样后的多数类样本与少数类样本组成临时训练集 D_t，归一化 W_t；
	在 D_t 上根据 W_t 训练 $h_t(x)$，计算分类误差 $e_t=\sum_{i=1}^{n} w_{ti} I(h_t(x_i)\neq y_i)$；
	计算权重 $\alpha_t=0.5\ln((1-e_t)/e_t)$ 和权重因子 $\beta_i=-0.5y_i h_t(x_i)c_i+0.5$；
	更新样本的权重：$W_{t+1}=W_t(i)\exp(-\alpha_t y_i h_t(x_i)\beta_i)/Z_t$，$Z_t$ 为归一化因子；
	End For

步骤 3	组合基分类器 $H(x)=\text{sign}(\sum_{t=1}^{T}\alpha_t h_t(x_i))$。

4.3 算法复杂度分析

令 n 为输入数据的样本数量，t 为迭代次数，f 为弱分类器的时间复杂度。在 DPBCPUSBoost 中，计算误分代价矩阵 $\boldsymbol{C}$ 的时间复杂度为 $O(n^2)$，计算采样权重 η' 的时间复杂度为 $O(n^2)$，欠采样的时间复杂度为 $O(n)$，则 DPBCPUSBoost 的整体时间复杂度为 $O(n^2+tn+tf)$。AdaBoost 的时间复杂度为 $O(n+tf)$，AdaCost 的时间复杂度为 $O(n+tf)$，RUSBoost 的时间复杂度为 $O(n+tn+tf)$，USCBoost 的时间复杂度为 $O(n+tn+tf)$。

令 h 为弱分类器的空间复杂度。DPBCPUSBoost 的空间复杂度主要在于计算密度峰值，其整体复杂度为 $O(n^2+h)$。AdaBoost 的空间复杂度为 $O(n+h)$，AdaCost 的空间复杂度为 $O(n+h)$，RUSBoost 的空间复杂度为 $O(n+h)$，USCBoost 的空间复杂度为 $O(n+h)$。

综上所述，相较于采用欠采样方法的 RUSBoost 和 USCBoost 算法，DPBCPUSBoost 的时间复杂度和空间复杂度更高。但是，对于没有采用欠采样方法的 AdaBoost 和 AdaCost 算法，由于它们在训练弱分类器时输入了全部样本，而 DPBCPUSBoost 只输入欠采样后的小部分样本，所以相较于上述两个算法，DPBCPUSBoost 的训练过程消耗了更少的时间和空间。

4.4 实验结果与分析

4.4.1 实验设置

本节设计了多个对比实验来验证提出的 DPBCPUSBoost 的有效性，实验环境与 3.4 节中的相同。

实验选取 AdaBoost、AdaCost、RUSBoost 和 USCBoost 作为对比算法：AdaBoost 是一个经典的集成算法，它将多个弱分类器级联，具有泛化性能好、分类精度高等优点；AdaCost 在 AdaBoost 的基础上，将代价调整函数添加到样本

权重更新步骤中，可以更快地增加高误分代价样本的权重；RUSBoost 在 AdaBoost 训练弱分类器前，先对多数类样本进行随机欠采样，使数据分布更加平衡，而且能提高运行效率；USCBoost 结合了 AdaCost 的代价调整函数和 RUSBoost 的随机欠采样方法，在初次迭代欠采样后，根据样本权重对多数类样本进行欠采样，以使分类器着重关注被误分的少数类样本。本文提出的算法 DPBCPUSBoost 在 USCBoost 的基础上，采用了新的欠采样方法和误分代价计算方法。因此，将上述 4 个算法与 DPBCPUSBoost 进行对比，能够验证本文所提出算法的有效性。

对比实验中各算法参数设置如下：算法迭代次数均设置为 10，其中 DPBCPUSBoost 的截断距离截取阈值参数α设置为 2，AdaCost 和 USCBoost 的误分代价计算方式均采用类依赖代价形式。

对比实验中各算法的弱分类器为决策树桩(Decision Stump)，因为 Decision Stump 是单层决策树，它的结构非常简单，一般不单独作为分类器，但是常作为 Boosting 算法中的弱分类器。

4.4.2 实验数据集

对比实验基于 10 个 KEEL 不平衡数据集[①]，表 4.3 展示了数据集的相关信息，其中不平衡比例为多数类样本数量与少数类样本数量的比值。

KEEL 不平衡数据集是由原始数据集转换后得到的二类别不平衡数据集，例如在 vehicle3 数据集中，少数类样本是原 vehicle 中的类别 3 构成，多数类样本则是由剩余类别组成，在 ecoli-0-6-7_vs_5 数据中，少数类样本是由类别 0、6 和 7 组成，多数类由类别 5 组成。实验使用的数据集均已使用五折划分成了 5 个子数据集。

4.4.3 不平衡数据分类性能实验

为验证 DPBCPUSBoost 的有效性，对 5 个算法的不平衡数据分类性能进行测试。图 4.2 展示了各算法取得最高性能的数据集数量，表 4.4 为 AdaBoost 和 AdaCost 的分类性能测试结果，表 4.5-4.7 为 RUSBoost、USCBoost 和

[①] https://sci2s.ugr.es/keel/imbalanced.php

DPBCPUSBoost 的分类性能测试结果。

表 4.3 实验数据集

数据集（标记）	样本数	属性数	不平衡比例
vehicle3(D1)	846	18	2.52
ecoli2(D2)	336	7	5.46
yeast_0_5_6_7_9_vs_4(D3)	528	8	9.35
ecoli-0-6-7_vs_5(D4)	220	6	10.00
yeast_1_vs_7(D5)	459	7	13.87
yeast_1_4_5_8_vs_7(D6)	693	8	22.10
yeast_2_vs_8(D7)	482	8	23.10
yeast6(D8)	1484	8	39.15
poker-8-9_vs_6-5-5tst(D9)	1485	10	58.40
abalone19(D10)	4714	8	128.87

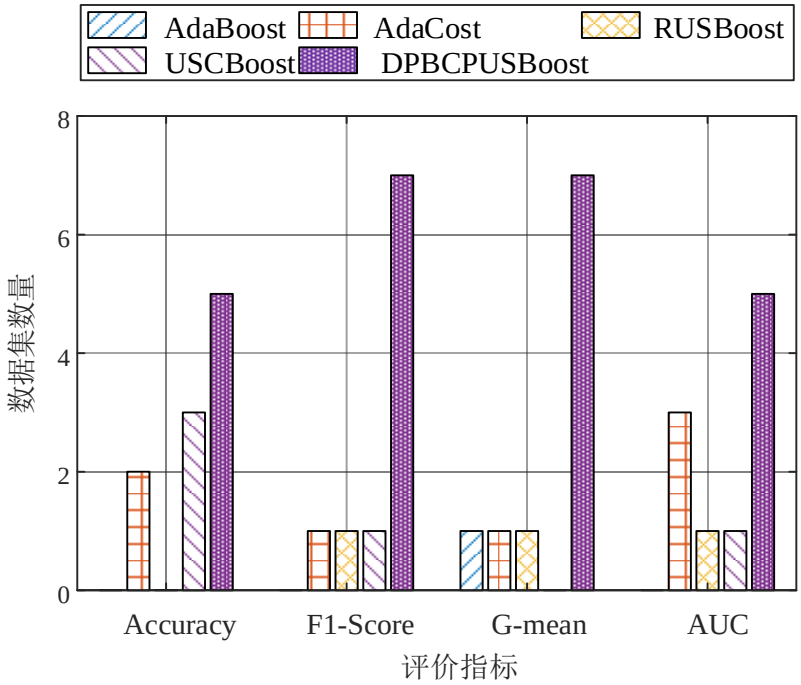


图 4.2 各算法取得最高性能的数据集数量

表 4.4 AdaBoost 和 AdaCost 的分类性能测试结果(%)

数据集	AdaBoost			
	Accuracy	F1-Score	G-mean	AUC
D1	50.71	50.83	50.94	61.96
D2	70.34	74.49	79.15	93.51
D3	66.84	70.09	73.66	72.53
D4	67.44	70.99	74.95	95.00
D5	60.60	65.60	71.50	81.76
D6	47.81	32.35	24.44	66.92
D7	53.26	57.70	62.94	61.68
D8	61.63	67.71	75.11	86.55
D9	49.08	32.92	24.77	42.12
D10	49.64	33.17	24.91	51.59
数据集	AdaCost			
	Accuracy	F1-Score	G-mean	AUC
D1	37.50	27.27	21.43	69.78
D2	83.69	85.20	86.76	92.63
D3	69.10	72.03	75.23	73.16
D4	65.38	71.77	79.55	96.25
D5	55.80	60.04	64.98	77.06
D6	65.65	55.04	60.92	68.83
D7	65.05	69.76	75.19	52.45
D8	56.95	64.94	75.53	92.34
D9	49.08	32.92	24.77	28.01
D10	50.31	54.41	59.25	60.17

由图 4.2 可知：AdaCost 的分类性能总体上优于 AdaBoost，因为其在样本权重调整过程中，进一步增加了高误分代价样本的权重；USCBoost 的性能总体上优于 RUSBoost，因为其在欠采样过程中保留了较高权重的样本。

表 4.5 RUSBoost 的分类性能测试结果

数据集	初次迭代性能				10 次迭代性能			
	Accuracy	F1-Score	G-mean	AUC	Accuracy	F1-Score	G-mean	AUC
D1	57.73	58.70	59.72	59.89	55.17	55.86	56.59	58.86
D2	66.03	70.46	75.55	96.22	85.82	87.18	88.62	96.84
D3	65.86	70.36	75.55	95.90	63.88	67.20	70.90	81.45
D4	57.74	61.36	67.39	97.50	73.31	72.30	72.80	93.75
D5	56.85	60.31	64.29	91.37	59.69	64.53	70.23	69.25
D6	53.17	56.80	61.39	86.95	51.83	55.73	60.80	76.19
D7	54.40	44.00	45.83	46.31	53.32	57.48	63.24	56.20
D8	57.75	65.52	75.72	85.08	59.88	66.67	75.25	84.66
D9	65.47	33.80	49.77	60.81	50.54	54.27	60.29	50.75
D10	51.38	56.08	66.61	53.38	50.22	54.48	60.50	38.97

表 4.6 USCBoost 的分类性能测试结果

数据集	初次迭代性能				10 次迭代性能			
	Accuracy	F1-Score	G-mean	AUC	Accuracy	F1-Score	G-mean	AUC
D1	57.41	58.39	59.40	59.55	51.21	51.02	50.86	55.26
D2	65.97	70.42	75.51	96.45	87.96	88.80	89.65	98.21
D3	65.96	70.44	75.59	95.85	60.98	64.72	68.94	73.89
D4	58.74	63.85	70.91	97.03	69.21	74.30	80.22	93.88
D5	58.43	62.02	66.17	91.37	61.48	65.94	71.10	81.92
D6	52.54	55.47	59.41	87.35	47.58	32.24	24.38	62.20
D7	56.31	44.75	48.19	48.78	56.16	52.93	53.24	62.55
D8	57.77	65.57	75.82	84.90	63.95	70.45	78.44	94.86
D9	68.33	30.40	49.19	61.53	58.96	32.94	29.82	40.81
D10	51.32	54.77	64.05	50.93	50.64	56.87	64.84	58.74

表 4.7 DPBCPUSBoost 的分类性能测试结果

数据集	初次迭代性能				10 次迭代性能			
	Accuracy	F1-Score	G-mean	AUC	Accuracy	F1-Score	G-mean	AUC
D1	57.24	58.24	59.27	59.34	63.30	63.99	64.70	63.78
D2	65.46	70.24	75.78	96.99	90.38	91.17	91.96	99.23
D3	65.36	69.99	75.34	95.75	75.73	77.77	79.94	75.68
D4	60.14	66.54	74.60	97.55	73.68	76.96	80.54	93.75
D5	56.84	60.63	65.12	90.57	52.51	55.10	57.97	84.31
D6	53.39	57.22	61.99	88.81	65.15	69.93	75.46	83.71
D7	57.89	39.86	45.60	47.25	47.89	32.38	24.46	82.07
D8	57.44	65.63	75.68	84.76	64.85	70.99	78.45	93.72
D9	67.40	31.82	49.81	61.61	49.03	32.90	24.76	54.26
D10	50.43	57.75	68.23	53.60	50.37	60.30	75.09	32.61

在初次迭代时，DPBCPUSBoost 使用 DPBCPUS 方法对多数类样本进行欠采样，而 RUSBoost 和 USCBoost 使用 RUS 方法对多数类样本进行欠采样。因此，为验证在初次迭代时应用 DPBCPUS 方法的有效性，对这 3 个算法在不同数据集上初次迭代和 10 次迭代后的分类性能进行测试，结果如表 4.5-4.7 所示。

由表 4.5-4.7 中初次迭代性能测试结果可以得出结论：相较于 RUS 方法，DPBCPUS 方法具有一定的优势，这是因为它尽量保留了决策边界区域内易误分的多数类样本，能够使分类器更加关注这些区域内的样本，但是在一些类重叠度较高的数据集上，该方法可能会降低分类性能。

由图 4.2 和表 4.5-4.7 的实验结果可以得出以下两个结论：

（1）DPBCPUSBoost 在 AUC、G-mean 和 F1-Score 指标上获得最高性能的数据集数量分别为 5 个、5 个和 4 个，均多于 RUSBoost 和 USCBoost。

（2）DPBCPUSBoost 在 Accuracy 指标上获得最高性能的数据集数量为 3 个，要少于 RUSBoost，因为 Accuracy 指标易受多数类样本数量影响，例如在不平衡比较高的 abalone19(D10)数据集上，RUSBoost 在 AUC、G-mean 和 F1-Score 指标上的表现均要差于 DPBCPUSBoost。

由以上实验结果可以得出结论：DPBCPUSBoost 在 Accuracy、F1-Score、G-mean 和 AUC 这 4 个评价指标上，获得最高性能的数据集数量分别为 5 个、7 个、7 个和 5 个，总体上 DPBCPUSBoost 的分类性能优于其他 4 个算法，这是因为 DPBCPUSBoost 考虑了样本的类内和类间分布情况，采用了新的欠采样方法和误分代价定义方法，尽量保留了类簇中有价值的信息。

图 4.3 展示了 5 个算法在 10 个数据集上的 50 次平均运行时间。

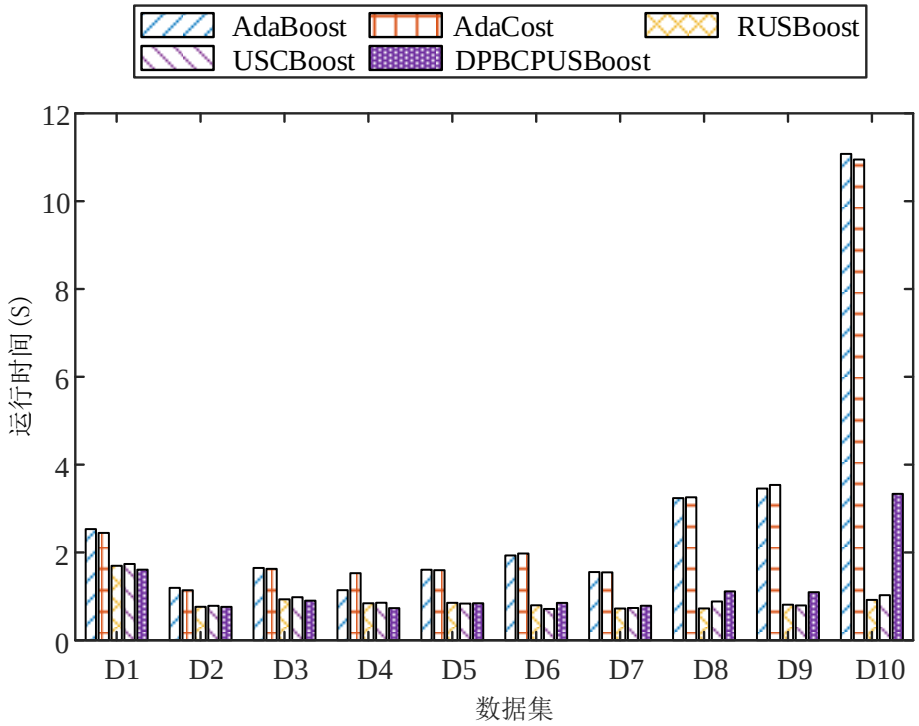


图 4.3 不同算法在各个数据集上的运行时间

由图 4.3 可知：AdaBoost 和 AdaCost 的运行时间显著多于 RUSBoost、USCBoost 和 DPBCPUSBoost，这是因为 AdaBoost 和 AdaCost 没有对多数类样本进行欠采样，训练集中存在更多的样本，因此训练时间会更长；DPBCPUSBoost 的运行时间要多于 RUSBoost 和 USCBoost，这是因为密度峰值计算需要更多时间。因此，DPBCPUSBoost 利用密度峰值信息定义采样权重和误分代价，牺牲了算法运行效率，但是提升了不平衡数据分类性能。

4.5 本章小结

针对目前不平衡数据分类算法中存在的一些问题，本文提出了一种密度峰值优化的球簇划分欠采样不平衡数据分类算法(DPBCPUSBoost)。首先根据密度峰值信息，使有更高代表性的样本具有更高的误分代价。然后基于“球簇划分”思想，将多数类球簇划分出“易误分区域”，并增加此区域内样本的采样权重，使决策边界区域内和位于局部中心的多数类样本获得更多关注。基于 10 个基准数据集上的对比实验结果表明：密度峰值能够用于发现具有较高价值的样本，DPBCPUSBoost 利用密度峰值优化欠采样和误分代价计算方法是有效的，相较于对比算法能够获得更高的不平衡数据分类性能。

5 总结与展望

5.1 全文总结

针对目前半监督自训练算法和不平衡数据分类算法研究中存在的一些问题，本文提出了两种利用密度峰值进行优化的改进算法：

（1）STDPM

针对自训练算法中高置信度样本选取问题，本文提出了一种密度峰值隶属度优化的自训练算法(STDPM)。STDPM 的主要创新点如下：基于密度峰值，提出了样本原型的定义，样本的更高密度最近邻为其“原型”，根据样本与其原型之间的层次关系构造了一种结构“原型树”，基于原型树定义了样本的“近亲结点”；基于有标签样本的类别信息和无标签样本的原型，充分考虑数据的局部和全局结构，定义了新的类簇隶属度“密度峰值隶属度”；将原型树中搜索出的有标签样本的无标签近亲结点作为潜在高置信度样本，其中隶属度大于设定阈值的样本作为高置信度样本被用于扩充训练集。经多个实验证明，STDPM 利用原型树和近亲结点能够提升选取高置信度样本的速度，利用密度峰值隶属度能够提升选取高置信度样本的质量。

（2）DPBCPUSBoost

不平衡数据分类面临多数类样本信息损失和少数类分类精度不高等问题，针对以上问题，本文提出了一种密度峰值优化的球簇划分欠采样不平衡数据分类算法(DPBCPUSBoost)。DPBCPUSBoost 的主要创新点如下：提出了一种密度峰值优化的球簇划分欠采样方法(DPBCPUS)，首先对密度和峰值都较大的样本赋予更大的采样权重，然后增大多数类球簇中“易误分区域”内样本的采样权重，尽量保留决策边界区域内的样本，最后根据采样权重对多数类样本进行欠采样；提出了一种兼顾类依赖和样本依赖的误分代价计算方法，首先依据类别分布信息赋予少数类样本更高的误分代价，然后提高密度和峰值都较大的局部中心样本的误分代价，最后通过代价调整函数更新样本权重。经多个实验证明，DPBCPUSBoost 利用密度峰值信息能够发现数据中具有更高价值的样本，并给予它们更高的关注度，从而提升分类精度。

5.2 研究展望

本文提出了一种半监督自训练算法 STDPM 和一种不平衡数据分类算法 DPBCPUSBoost。本文提出算法的分类性能相较于对比算法有一定的提升，但是还存在一些需要继续改进的问题：

（1）高维数据的距离计算

STDPM 和 DPBCPUSBoost 都需要计算数据的局部密度，然而密度计算依赖于距离计算，在对高维数据进行距离计算时会产生“维度灾难”问题，本文采用的相似性度量方式“欧式距离”将无法准确反映数据之间的关系。因此，在今后的研究工作中，需要寻找一些更适合高维数据的距离计算方法。

（2）密度不均匀数据的局部密度计算

对密度不均匀的数据进行分类存在以下问题：利用密度峰值发现不了准确的类簇局部中心点，从而会影响采样权重和误分代价的计算；利用密度峰值构造的原型树也不能很好地反映数据的层次结构，从而会导致高置信度样本选取不准确。未来将着重改进密度峰值的计算方法，以增强对密度不均匀数据的适用性。

（3）算法时间复杂度

密度峰值计算的时间复杂度为$O(n^2)$，在较大规模的数据中，算法的运行效率将会成为瓶颈。未来将减少密度峰值中的距离计算量作为工作重点，寻求降低算法时间复杂度的方法。

（4）类重叠数据分类

DPBCPUS 方法需要对数据进行“球簇划分”，然而对于一些类间重叠度较高的数据，“球簇划分”方法不能很好地区分不同类的样本。未来将设计一种适用复杂数据的类簇划分方法，以解决类重叠的不平衡数据的分类问题。

参考文献

[1] Sen P C, Hajra M, Ghosh M. Supervised classification algorithms in machine learning: A survey and review[M]. Singapore: Springer, 2020: 99-111.

[2] Alloghani M, Al-Jumeily D, Mustafina J, et al. A systematic review on supervised and unsupervised machine learning algorithms for data science[J]. Supervised and Unsupervised Learning for Data Science, 2020: 3-21.

[3] Chong Y, Ding Y, Yan Q, et al. Graph-based semi-supervised learning: A review[J]. Neurocomputing, 2020, 408.

[4] Kostopoulos G, Karlos S, Kotsiantis S, et al. Semi-supervised regression: A recent review[J]. Journal of Intelligent & Fuzzy Systems, 2018, 35(2).

[5] 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(06): 19-27.

[6] Zhu X J. Semi-supervised learning literature survey[J]. world, 2005, 10: 10.

[7] Li M, Zhou Z-H. SETRED: Self-training with editing[C]//Pacific-Asia Conference on Knowledge Discovery and Data Mining. Berlin, Heidelberg: Springer, 2005: 611-621.

[8] Zighed D A, Lallich S, Muhlenbach F. Separability index in supervised learning[C]//European Conference on Principles of Data Mining and Knowledge Discovery. Berlin, Heidelberg: Springer, 2002: 475-487.

[9] Muhlenbach F, Lallich S, Zighed D A. Identifying and handling mislabelled instances[J]. Journal of Intelligent Information Systems, 2004, 22(1): 89-109.

[10] Xia S, Peng D, Meng D, et al. Ball k-Means: Fast Adaptive Clustering With No Bounds[J]. IEEE transactions on pattern analysis and machine intelligence, 2022, 44(1): 87-99.

[11] Hasecke F, Hahn L, Kummert A. Fast lidar clustering by density and connectivity[J]. arXiv preprint arXiv:2003.00575, 2020.

[12] Gan H, Sang N, Huang R, et al. Using clustering analysis to improve semi-supervised classification[J]. Neurocomputing, 2013, 101: 290-298.

[13] Gan H, Tong X, Jiang Q, et al. Discussion of FCM algorithm with partial supervision[C]//Proceedings of the Eighth International Symposium on Distributed Computing and Applications to Business, Engineering and Science. Beijing, China: Publishing House of Electronics Industry, 2009: 27-31.

[14] Tong X, Jiang Q, Sang N, et al. The feature weighted FCM algorithm with semi-supervised[C]//Proceedings of the Eighth International Symposium on Distributed Computing and Applications to Business, Engineering and Science. Beijing, China: Publishing House of Electronics Industry, 2009: 22-26.

[15] Rodriguez A, Laio A. Clustering by fast search and find of density peaks[J]. Science, 2014, 344(6191): 1492-1496.

[16] 陈叶旺, 申莲莲, 钟才明等. 密度峰值聚类算法综述[J]. 计算机研究与发展, 2020, 57(02): 378-394.

[17] 丁志成, 葛洪伟. 优化分配策略的密度峰值聚类算法[J]. 计算机科学与探索, 2020, 14(05): 792-802.

[18] 丁世飞, 徐晓, 王艳茹. 基于不相似性度量优化的密度峰值聚类算法[J]. 软件学报, 2020, 31(11): 3321-3333.

[19] 柏锷湘, 罗可, 罗潇. 结合自然和共享最近邻的密度峰值聚类算法[J]. 计算机科学与探索, 2020: 1-13.

[20] 刘娟, 万静. 自然反向最近邻优化的密度峰值聚类算法[J]. 计算机科学与探索, 2020: 1-12.

[21] 金辉, 钱雪忠. 自然最近邻优化的密度峰值聚类算法[J]. 计算机科学与探索, 2019, 13(04): 711-720.

[22] 钱雪忠, 金辉. 自适应聚合策略优化的密度峰值聚类算法[J]. 计算机科学与探索, 2020, 14(04): 712-720.

[23] 刘沧生, 许青林. 基于密度峰值优化的模糊 C 均值聚类算法[J]. 计算机工程与应用, 2018, 54(14): 153-157.

[24] 谢娟英, 屈亚楠. 密度峰值优化初始中心的 K-medoids 聚类算法[J]. 计算机科学与探索, 2016, 10(02): 230-247.

[25] Wu D, Shang M, Luo X, et al. Self-training semi-supervised classification based on

density peaks of data[J]. Neurocomputing, 2018, 275: 180-191.

[26] 艾震鹏, 王振友. 基于数据密度的半监督自训练分类算法[J]. 计算机应用研究, 2019, 36(04): 1072-1074.

[27] 卫丹妮, 杨有龙, 仇海全. 结合密度峰值和切边权值的自训练算法[J]. 计算机工程与应用, 2020: 1-8.

[28] 卫丹妮. 基于切边权值统计的自训练分类研究[D]. 西安: 西安电子科技大学, 2020.

[29] 吕佳, 李婷婷. 半监督自训练方法综述[J]. 重庆师范大学学报(自然科学版), 2021, 38(05): 98-106.

[30] 黎隽男. 半监督自训练方法的研究[D]. 重庆: 重庆师范大学, 2018.

[31] Karlos S, Fazakis N, Panagopoulou A-P, et al. Locally application of naive Bayes for self-training[J]. Evolving Systems, 2017, 8(1): 3-18.

[32] Hady M F A, Schwenker F. Co-training by committee: a new semi-supervised learning framework[C]//2008 IEEE International Conference on Data Mining Workshops. Piscataway, NJ: IEEE, 2008: 563-572.

[33] Zhou Z-H, Li M. Tri-training: Exploiting unlabeled data using three classifiers[J]. IEEE transactions on knowledge and data engineering, 2005, 17(11): 1529-1541.

[34] Shi L, Ma X, Xi L, et al. Rough set and ensemble learning based semi-supervised algorithm for text classification[J]. Expert Systems with Applications, 2011, 38(5): 6300-6306.

[35] Livieris I E, Kanavos A, Tampakas V, et al. An Auto-Adjustable Semi-Supervised Self-Training Algorithm[J]. Algorithms, 2018, 11(9): 139.

[36] Zhou X, Hu Y, Liang W, et al. Variational LSTM enhanced anomaly detection for industrial big data[J]. IEEE Transactions on Industrial Informatics, 2020, 17(5): 3469-3477.

[37] 胡姣姣, 王晓峰, 张萌等. 基于深度学习的时间序列数据异常检测方法[J]. 信息与控制, 2019, 48(01): 1-8.

[38] 张跃飞, 王敬飞, 陈斌等. 基于改进的 Mask R-CNN 的公路裂缝检测算法[J]. 计算机应用, 2020, 40(S2): 162-165.

[39] Lin P, Ye K, Xu C-Z. Dynamic network anomaly detection system by using deep learning techniques[C]//CLOUD 2019: International Conference on Cloud Computing. Cham, Switzerland: Springer, 2019: 161-176.

[40] 刘颖, 杨轲. 基于深度集成学习的类极度不均衡数据信用欺诈检测算法[J]. 计算机研究与发展, 2021, 58(03): 539-547.

[41] Zhu H, Liu G, Zhou M, et al. Optimizing Weighted Extreme Learning Machines for imbalanced classification and application to credit card fraud detection[J]. Neurocomputing, 2020, 407: 50-62.

[42] Li Z, Huang M, Liu G, et al. A hybrid method with dynamic weighted entropy for handling the problem of class imbalance with overlap in credit card fraud detection[J]. Expert Systems with Applications, 2021, 175: 114750.

[43] 易东义, 邓根强, 董超雄等. 基于图卷积神经网络的医保欺诈检测算法[J]. 计算机应用, 2020, 40(05): 1272-1277.

[44] 刘枭, 王晓国. 基于密集子图的银行电信诈骗检测方法[J]. 计算机应用, 2019, 39(04): 1214-1219.

[45] Lu C, Lin S, Liu X, et al. Telecom Fraud Identification Based on ADASYN and Random Forest[C]//ICCCS 2020: 2020 5th International Conference on Computer and Communication Systems Piscataway, NJ: IEEE, 2020: 447-452.

[46] 王伟, 谢耀滨, 尹青. 针对不平衡数据的决策树改进方法[J]. 计算机应用, 2019, 39(03): 623-628.

[47] 徐玲玲, 迟冬祥. 面向不平衡数据集的机器学习分类策略[J]. 计算机工程与应用, 2020, 56(24): 12-27.

[48] 陈木生, 卢晓勇. 三种用于垃圾网页检测的随机欠采样集成分类器[J]. 计算机应用, 2017, 37(02): 535-539+558.

[49] 王俊红, 闫家荣. 基于欠采样和代价敏感的不平衡数据分类算法[J]. 计算机应用, 2021, 41(01): 48-52.

[50] Ersin K, Sedat K, Akif S M, et al. DEBOHID: A differential evolution based oversampling approach for highly imbalanced datasets[J]. Expert Systems with Applications, 2021, 169.

[51] Justin E, Stefan L. Conditional Wasserstein GAN-based Oversampling of Tabular Data for Imbalanced Learning[J]. Expert Systems with Applications, 2021, 174: 114582.

[52] Xinyue W, Jian X, Tieyong Z, et al. Local distribution-based adaptive minority oversampling for imbalanced data classification[J]. Neurocomputing, 2021, 422.

[53] Gao X, Ren B, Zhang H, et al. An ensemble imbalanced classification method based on model dynamic selection driven by data partition hybrid sampling[J]. Expert Systems with Applications, 2020, 160.

[54] Zhu Y, Yan Y, Zhang Y, et al. EHSO: Evolutionary Hybrid Sampling in overlapping scenarios for imbalanced learning[J]. Neurocomputing, 2020, 417: 333-346.

[55] Yen S-J, Lee Y-S. Cluster-based under-sampling approaches for imbalanced data distributions[J]. Expert Systems with Applications, 2009, 36(3): 5718-5727.

[56] Lin W-C, Tsai C-F, Hu Y-H, et al. Clustering-based undersampling in class-imbalanced data[J]. Information Sciences, 2017, 409: 17-26.

[57] Tsai C-F, Lin W-C, Hu Y-H, et al. Under-sampling class imbalanced datasets by combining clustering analysis and instance selection[J]. Information Sciences, 2019, 477: 47-54.

[58] Freund Y, Schapire R E. Experiments with a new boosting algorithm[C]//ICML 1996: Proceedings of the Thirteenth International Conference on International Conference on Machine Learning. San Francisco, CA: Morgan Kaufmann Publishers Inc, 1996: 148-156.

[59] Fan W, Stolfo S J, Zhang J, et al. AdaCost: Misclassification Cost-Sensitive Boosting[C]//ICML 1999: Proceedings of the Sixteenth International Conference on Machine Learning. San Francisco, CA: Morgan Kaufmann Publishers Inc., 1999: 97-105.

[60] Schölkopf B, Platt J C, Shawe-Taylor J, et al. Estimating the support of a high-dimensional distribution[J]. Neural computation, 2001, 13(7): 1443-1471.

[61] Fu J, Lee S. Certainty-based active learning for sampling imbalanced datasets[J]. Neurocomputing, 2013, 119: 350-358.

[62] Liu W, Chawla S, Cieslak D A, et al. A robust decision tree algorithm for imbalanced data sets[C]//Proceedings of the 2010 SIAM International Conference on Data Mining. SIAM, 2010: 766-777.

[63] Seiffert C, Khoshgoftaar T M, Van Hulse J, et al. RUSBoost: A hybrid approach to alleviating class imbalance[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2009, 40(1): 185-197.

[64] Liu X-Y, Wu J, Zhou Z-H. Exploratory undersampling for class-imbalance learning[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 2008, 39(2): 539-550.

[65] Vens C, Struyf J, Schietgat L, et al. Decision trees for hierarchical multi-label classification[J]. Machine learning, 2008, 73(2): 185-214.

[66] Zhang M-L, Zhou Z-H. ML-KNN: A lazy learning approach to multi-label learning[J]. Pattern Recognition, 2007, 40(7): 2038-2048.

[67] Joachims T. Optimizing search engines using clickthrough data[C]//Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. New York, NY: Association for Computing Machinery, 2002: 133-142.

[68] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: synthetic minority over-sampling technique[J]. Journal of artificial intelligence research, 2002, 16: 321-357.

[69] Chawla N V, Lazarevic A, Hall L O, et al. SMOTEBoost: Improving prediction of the minority class in boosting[C]//European conference on principles of data mining and knowledge discovery. Berlin, Heidelberg: Springer, 2003: 107-119.

[70] Platt J. Sequential Minimal Optimization : A Fast Algorithm for Training Support Vector Machines[J]. Microsoft Research Technical Report, 1998, MSR-TR-98-14.

[71] 游文霞, 申坤, 杨楠等. 基于 AdaBoost 集成学习的窃电检测研究[J]. 电力系统保护与控制, 2020, 48(19): 151-159.

[72] Zhang T, Ramakrishnan R, Livny M. BIRCH: an efficient data clustering method for very large databases[J]. ACM sigmod record, 1996, 25(2): 103-114.

[73] Guha S, Rastogi R, Shim K. ROCK: A robust clustering algorithm for categorical attributes[J]. Information Systems, 2000, 25(5): 345-366.

[74] Guha S, Rastogi R, Shim K. CURE: An efficient clustering algorithm for large databases[J]. ACM sigmod record, 1998, 27(2): 73-84.

[75] Karypis G, Han E-H, Kumar V. Chameleon: Hierarchical clustering using dynamic modeling[J]. Computer, 1999, 32(8): 68-75.

[76] 王哲川. 自动确定类数的密度峰值聚类算法研究[D]. 西安: 西安电子科技大学, 2020.

[77] 张立宁. 改进的密度峰值聚类算法研究[D]. 西安: 西安电子科技大学, 2019.

[78] 晏焕钱. 基于密度峰值聚类的两种改进算法的研究[D]. 兰州: 兰州大学, 2018.

[79] 刘建伟, 刘媛, 罗雄麟. 半监督学习方法[J]. 计算机学报, 2015, 38(08): 1592-1617.

[80] 江兵兵. 基于贝叶斯方法的半监督学习算法研究[D]. 合肥: 中国科学技术大学, 2019.

[81] 李宁宁. 基于半监督协同训练的文本情感分类研究[D]. 合肥: 合肥工业大学, 2015.

[82] 赵琴琴. 多视图学习方法研究[D]. 西安: 西安电子科技大学, 2017.

[83] 赵志凯. 半监督学习及其在煤矿瓦斯安全信息处理中的应用研究[D]. 徐州: 中国矿业大学, 2012.

[84] 王秀秀. 基于稀疏图的半监督学习方法研究[D]. 西安: 西安电子科技大学, 2013.

[85] Belkin M, Niyogi P, Sindhwani V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples[J]. Journal of machine learning research, 2006, 7(11).

硕士期间的研究成果

1. 发表学术论文

[1] **刘学文**, 王继奎, 杨正国, 易纪海, 李冰, 聂飞平. 近亲结点图编辑的Self-Training算法[J/OL]. 计算机工程与应用, 2021: 1-14 [2021-03-11]. http://kns.cnki.net/kcms/detail/11.2127.tp.20210331.1557.036.html.（已录用）

[2] **刘学文**, 王继奎, 杨正国, 李冰, 聂飞平. 密度峰值隶属度优化的半监督 Self-Training 算法[J/OL]. 计算机科学与探索, 2021: 1-13 [2021-04-19]. http://kns.cnki.net/kcms/detail/11.5602.tp.20210415.1147.006.html. （已录用）

[3] **刘学文**, 王继奎, 杨正国, 李强, 易纪海, 李冰, 聂飞平. 密度峰值优化的球簇划分欠采样不平衡数据分类算法[J/OL]. 计算机应用, 2022: 1-10 [2022-03-14]. http://kns.cnki.net/kcms/detail/51.1307.tp.20220310.1720.009.html. （已录用）

[4] Wang J K, Yang Z G, **Liu X W**, Li B, Yi J H, Nie F P. Projected Fuzzy C-Means with Probabilistic Neighbors[J]. Information Sciences, 2022.（已录用）

[5] 王继奎, 杨正国, **刘学文**, 易纪海, 李冰, 聂飞平. 一种基于极大熵的快速无监督线性降维方法[J]. 软件学报, 2021.（已录用）

[6] Yang Z G, Wang J K, Li Q, Yi J H, **Liu X W**, Nie F P. Graph optimization for unsupervised dimensionality reduction with probabilistic neighbors[J]. Applied Intelligence, 2022: 1-14.

[7] 王继奎, 杨正国, 易纪海, **刘学文**, 王会勇, 聂飞平. 稀疏约束的嵌入式模糊均值聚类算法[J]. 复旦学报(自然科学版), 2020, 59(06): 725-733.

2. 参与科研项目

[1] 甘肃高等学校创新能力提升项目. 基于原型树的无监督学习方法与大数据应用研究(2019B-97). 2021.12.

[2] 兰州财经大学重点项目. 统一的降维和聚类学习模型与算法研究(Lzufe2020B-010). 2021.10.

[3] 甘肃高等学校创新能力提升项目. 基于万物互联的智慧交通设计与技术研究(2019A-069). 2021.12.

后 记

韶华易逝，流年似水，不经意间我的硕士生涯就要结束了。在此，我要特别感谢那些三年来给予我帮助和关心的老师、同学和朋友，因为有他们的指引和陪伴，我这三年过得非常充实和快乐。

首先，我要感谢学校兢兢业业的老师和工作人员。疫情暴发两年来，我们能够健康地在学校生活和学习，都要归功于那些在我们身后默默付出的老师和工作人员，感谢他们给我们创造了一个安全舒适的生活和学习环境。

然后，我要感谢实验室团队的老师们，特别是我的指导老师聂老师和王老师。聂老师在人工智能领域有着深厚的知识积淀，对前沿方向把握精准，正是因为有他的指引，我才能在科研路上少走很多弯路。刚踏入科研这条路时我懵懂无知，实验不会做、论文不会写，是王老师一遍一遍耐心指导，我的学术科研能力才得以迅速提高。年轻的我做事总是莽莽撞撞，不够细心和成熟。非常感谢王老师在这三年里包容我，并教会我很多做事和做人的道理。我还要感谢实验室团队里的杨老师、易老师和武老师，他们在专业知识学习上给了我非常多的帮助。

我还要感谢我的师弟师妹以及室友们，感谢他们在我亟须帮助的时候尽心尽力，感谢他们陪伴我度过这段短暂又难忘的时光！我还要感谢我的高中和大学好友，非常感谢他们与我分享一些工作和生活中的宝贵经验。

我还要衷心感谢我的父母。父母含辛茹苦养育我二十多年，时刻关心我的生活和学习状况，自从上学后我却鲜有时间去陪伴他们。二十多年来他们一直承担我的学费和生活费，但如今我还没有太多机会来报答他们的养育之恩。

最后，我还要特别感谢所有参与审稿和答辩的老师们，感谢你们在百忙之中提出专业意见和建议!