分类问题作业报告

Ziheng Wang
wangziheng@buaa.edu.cn

Abstract

本次作业报告旨在巩固和应用课上学习的决策树、集成学习和核方法的相关知识，重点通过Decision Trees、AdaBoost + DecisionTrees和SVM三种方法对一个3D数据集进行分类，其中尝试了SVM三种不同核在该任务中的表现。在复习相关内容的过程中，加深了对于课程内容的理解，增强了根据不同任务选择合适分类算法的能力。

Introduction

决策树是一种基于树形结构的监督学习算法，常用于分类或回归任务。它通过递归分裂数据集，根据特征划分形成一系列“决策节点”和“叶节点”，最终构建出树形模型。由于其结构直观、易于理解、无需复杂的数据预处理，且能够处理非线性关系，决策树训练速度较快，适合处理小到中等规模的数据集。然而，它容易发生过拟合，并且对数据分布较为敏感。AdaBoost是一种集成学习方法，通过将多个弱学习器（如浅层决策树）结合，形成一个强学习器。它具有较好的鲁棒性和泛化能力，但对异常值敏感，且训练过程较为缓慢。SVM是一种基于几何直观的监督学习算法，主要用于分类和回归任务。其目标是寻找一个超平面，使得不同类别的样本间隔最大化。借助核技巧，SVM可以有效避免维度灾难，因此在高维数据上表现较好，但计算复杂度高，且参数选择较为复杂。

在本次作业中，采用了Decision Trees、AdaBoost + DecisionTrees和SVM三种方法来对给定的高维数据集进行二分类，其中SVM算法使用了三种不同的核函数，并比较了它们的分类性能。由于数据为三维数据，选择合适核函数的SVM算法可能在此任务中表现最佳。

Methodology

本部分将详细阐述数据分类任务中所用模型的原理，包括具体的实现步骤和过程。

M1: Decision Trees

决策树是一种基本的分类与回归方法，其本质是将数据空间通过一系列条件判断划分成多个子空间，从而达到对样本进行分类或预测的目的。决策树的构建过程可以理解为在特征空间中构建一个树状结构，其中每一个非叶子节点代表一次特征判断，每一个叶子节点代表一个最终的分类结果或预测值。在数学原理上，决策树主要依赖于“贪心算法”来进行特征选择和划分，即每次选择当前最优的特征划分数据，使得子集在目标属性上的“纯度”最大化。衡量“纯度”的常见指标包括信息熵（entropy）、信息增益（information gain）以及基尼指数（Gini index）。以信息熵为例，它来源于信息论，用于衡量样本集合的不确定性。设有一个样本集合D，其包含K个类别，其中第k类样本所占比例为p_k，则集合的熵定义为：

$$Ent(D) = -\sum_{k=1}^{K} p_k \log_2 p_k$$

信息增益是某一特征对集合熵的减少量，即选择该特征进行划分后，样本集合的不确定性降低了多少。用数学表达为：

$$Gain(D,\ A) = Ent(D) - \sum_{v \in \mathrm{Values}(A)} \frac{|D^v|}{|D|} Ent(D^v)$$

其中，A 是待选的划分特征，$\mathrm{Values}(A)$表示特征A的所有可能取值，D^v表示在特征A上取值为v的子集。算法每次选择信息增益最大的特征进行划分，从而逐层构建决策树。另一种常用的纯度指标是基尼指数，它用于衡量在某一数据集中，任意两个样本被错误分类的概率，其定义为:

$$Gini(D) = 1 - \sum_{k=1}^{K} p_k^2$$

当基尼值越小，说明数据越“纯”，因此构建决策树时也可以选择使基尼指数最小化的特征进行划分。在实际应用中，ID3 算法使用信息增益，C4.5 使用信息增益率，CART（分类回归树）则使用基尼指数。

决策树的生成通常采用递归的方式进行，直到满足设定的停止条件为止，如达到最大深度、所有样本属于同一类或信息增益小于某个阈值。生成后的树可能过于复杂而导致过拟合，因此常常需要进行剪枝操作，分为预剪和后剪枝。

个人编写的程序中利用DecisionTreeClassifier()函数进行分类树模型的建立，采用两种衡量“纯度”的标准来训练模型，分别为信息熵（entropy）和基尼指数（Gini index），其余参数选取默认值（实际验证发现默认值准确率已经很高）。

M2: AdaBoost + DecisionTrees

Adaboost（Adaptive Boosting，自适应提升算法）是一种典型的集成学习方法，其基本思想是通过组合多个性能较弱的“弱分类器”来构造一个强分类器，从而提升整体的分类性能。在 Adaboost 框架中，最常用的弱分类器是决策树，特别是深度较小的决策树（如“决策桩”或限定最大深度的决策树），因为它们学习能力有限，容易受到样本分布的影响，而这恰恰符合 Adaboost 的设计理念。Adaboost 的数学原理可以理解为一个加权迭代优化的过程：假设训练数据集为$\{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_m, y_m)\}$，其中$x_i$为输入特征，$y_i \in \{-1, +1\}$为对应的标签。算法开始时为每个样本分配一个相等的权重$D_1(i) = \frac{1}{m}$。然后在每一轮$t = 1, \cdots, T$中，训练一个弱分类器$h_t(\boldsymbol{x})$，使其在当前加权样本分布下的分类误差最小，即计算误差率$\varepsilon_t = \sum_{i=1}^{m} D_t(i) \cdot \mathbb{I}(h_t(\boldsymbol{x}_i) \neq y_i)$。接着根据$\varepsilon_t$计算该弱分类器的权重$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$，并更新每个样本的权重，使分类错误的样本权重增加，分类正确的样本权重减小，其更新公式为：

$$D_{t+1}(i) = \frac{D_t(i) \cdot e^{-\alpha_t y_i h_t(\boldsymbol{x}_i)}}{Z_t}$$

其中Z_t是归一化因子，确保D_{t+1}是一个概率分布。这个过程不断迭代，使模型越来越关注那些难以分类的样本。最终，Adaboost 模型的输出为所有弱分类器按权重加权的结果：

$$H(\boldsymbol{x}) = \text{sign}\left(\sum_{t=1}^{T} \alpha_T h_t(\boldsymbol{x})\right)$$

这种加权投票机制使得 Adaboost 在实际应用中具有较强的鲁棒性和较低的泛化误差。

个人编写的代码中，调用AdaBoostClassifier来建立相应模型，同时经过实验比较，放弃默认SAMMA’R算法，采用SAMMA算法。同时选择训练50个弱分类器，由于每个弱分类器默认为单层决策桩，效果不理想，所以更改参数为max_depth=8的决策树。其余参数选择默认值。

M3: SVM

支持向量机（Support Vector Machine，简称 SVM）是一种典型的二类分类模型，其核心思想是通过构造一个最优超平面，在特征空间中将不同类别的样本分开，并使分类间隔最大化。数学上，SVM 本质上是一个凸优化问题，其目标是在满足分类正确的前提下，使得几何间隔最大。设训练集为$\{(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_m, y_m)\}$，其中$\boldsymbol{x}_i \in \mathbb{R}^n$是特征向量，

$y_i \in \{-1, +1\}$ 是标签，SVM 要求构造一个超平面 $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{b} = 0$，使得对所有样本满足 $y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + \boldsymbol{b}) \geq 1$，并且最大化间隔 $\frac{2}{\|w\|}$，从而转化为优化问题：

$$\min_{\boldsymbol{w},\boldsymbol{b}} \frac{1}{2}\|\boldsymbol{w}\|^2 \quad \text{s.t. } y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + \boldsymbol{b}) \geq 1, \ \forall i$$

这是硬间隔 SVM，当数据线性可分时适用。若数据存在噪声或不可完全分离，可引入松弛变量 ξ_i 得到软间隔 SVM，其目标函数变为：

$$\min_{\boldsymbol{w},\boldsymbol{b},\xi} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m}\xi_i \quad \text{s.t.} \quad y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + \boldsymbol{b}) \geq 1 - \xi_i, \ \xi_i \geq 0$$

其中 C 是惩罚系数，控制间隔最大化与分类错误之间的权衡。为了处理非线性可分问题，SVM 通过核函数将原始数据映射到高维特征空间，在高维空间中实现线性可分分类。常用核函数包括线性核 $K(\boldsymbol{x},\boldsymbol{x}') = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}'$，多项式核 $K(\boldsymbol{x},\boldsymbol{x}') = (\boldsymbol{x}^{\mathrm{T}}\boldsymbol{x}' + \boldsymbol{r})^d$，径向基函数核（RBF 核）$K(\boldsymbol{x},\boldsymbol{x}') = \exp\left(-\gamma\|\boldsymbol{x} - \boldsymbol{x}'\|^2\right)$，其中核函数的作用是通过内积运算隐式完成高维映射而不显式计算坐标，提高计算效率。在对偶问题中，支持向量由拉格朗日乘子 $\alpha_i > 0$ 对应的样本确定，它们位于间隔边界上或间隔错误区域，是最终决策函数的关键，决策函数可写为：

$$f(x) = \operatorname{sign}\left(\sum_{i=1}^{m}\alpha_i y_i K(\boldsymbol{x}_i,\boldsymbol{x}) + \boldsymbol{b}\right)$$

只有那些非零 α_i 对应的样本才参与预测，因此模型稀疏性好、计算高效。在实际应用中，SVM 对小样本、高维空间具有较强的分类能力，并通过选择合适的核函数与参数（如 γ、C、多项式核的 d）适应不同的分布结构。

本实验中调用SVC函数进行SVM模型建立，分别采用线性核（linear）、多项式核（poly）以及高斯核（rbf）来升维。其中多项式核选择degree=3的参数，其余核选择默认参数。

Experimental Studies

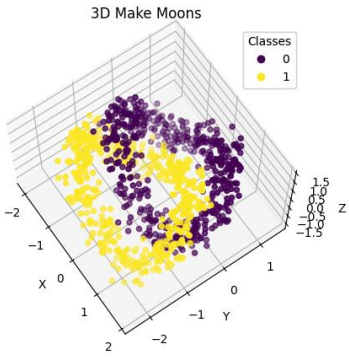
程序将三种不同方法分别在测试集上运行cycles次（该参数自己设计，本人设计为10），并将所得分类准确率取平均值，具体结果如表1所示。

表1 不同分类器测试集上表现

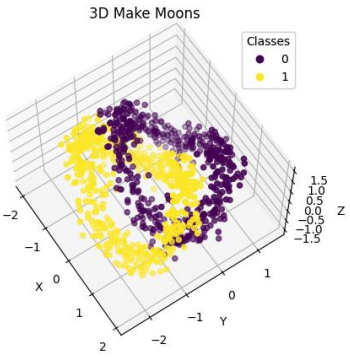
分类器	分类准确率
Decision Trees	0.965000
Adaboost+Decision Trees	0.983000
SVM with Linear Kernel	0.672000

SVM with Poly Kernel	0.863000
SVM with RBF Kernel	0.989000

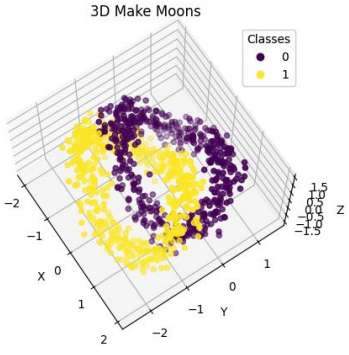
除此之外，本次实验还可视化了不同分类器的分类情况，并与测试集真实分类情况进行直观对比，如图1所示。通过观察可发现，测试集真实情况呈现3D月亮分布，有交叉，所以SVM模型用线性核去划分表现非常差。



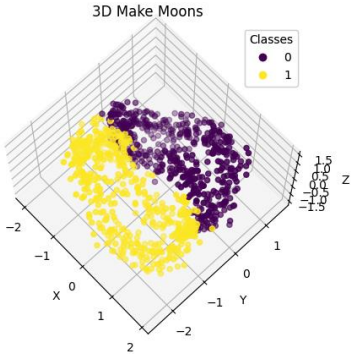
（a）测试集真实分类情况



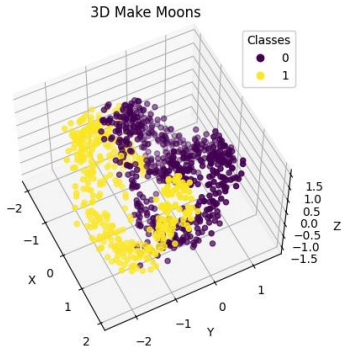
（b）决策树分类情况



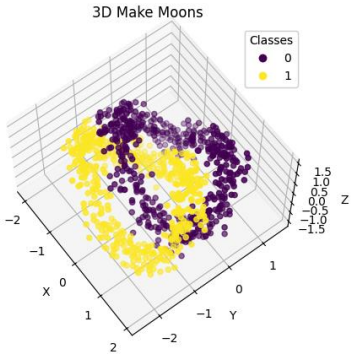
（c）Adamboost+决策树分类情况



（d）SVM 线性核分类情况



（e）SVM多项式核分类情况



（f）SVM高斯核分类情况

图 1　不同分类器分类情况可视化

下面我将进一步探讨不同模型在本任务中预测结果存在差异的原因。

首先，决策树和AdaBoost + 决策树的表现较好，且准确率较高。这与决策树本身的结构和工作原理密切相关。决策树通过递归地分割数据集，能够根据特征将数据有效划分成不同类别，且具有较强的可解释性。然而，决策树容易发生过拟合，特别是在数据集复杂或特征较多时，可能会在训练集上拟合过度，导致泛化能力下降。而在使用AdaBoost集成学习方法后，虽然个体弱分类器可能表现一般，但通过多轮训练和加权组合，以及将单层决策桩换为更深层的决策树，最终可以提升整体的分类性能，使得该方法对噪声和数据波动具有较强的鲁棒性。在本实验中，AdaBoost与决策树结合的模型表现优异，准确率为98.3%，相比单独使用决策树的表现有所提高。这表明，AdaBoost能够有效地克服单一决策树可能带来的过拟合问题，提升了分类器的泛化能力。

其次，SVM使用不同核函数的表现差异较大，使用线性核时，分类准确率仅为67.2%。这是因为该数据集的特点为3D月亮形，类别之间的边界是复杂的非线性关系，不可完全线性分割，需要进行升维分类。因此，SVM使用线性核时，无法找到合适的超平面进行划分，导致分类效果不理想。而在使用多项式核时，准确率提升至86.3%，这表明通过选择适当的核函数，SVM能够在高维空间中更好地进行分类。进一步实验发现，高斯核（RBF核）表现最为优异，准确率达到98.9%。这也因为RBF核通过将数据映射到更高维的特征空间，使得原本无法线性分割的样本在新空间中变得可分，进而有效提升了SVM的分类性能。

Conclusions

总的来说，模型的表现差异主要受数据本身的特性、模型的适应能力以及核函数选择等因素影响。通过这些实验结果，我深刻认识到在实际应用中，选择合适的模型和调参是提升分类性能的关键因素。