arXiv:2302.11573v1 [astro-ph.IM] 22 Feb 2023

运用机器学习技术分析天文数据

Mohammad H. Zhoolideh Haghighi*

Department of Physics, K.N. Toosi University of Technology, Tehran P.O. Box 15875-4416, Iran,
School of Astronomy, Institute for Research in Fundamental Sciences (IPM), Tehran, 19395-5746, Iran,

摘要

分类是机器学习（ML）和人工智能（AI）领域中一项常见任务，当输出为类别型变量时便会用到分类。现有多种模型试图根据观测值得出结论，因此分类算法会预测类别标签，并将其用于新数据的分类。逻辑回归、决策树、随机森林、支持向量机（SVM）、多层感知机、朴素贝叶斯和神经网络等常见分类模型，已被证明应用于许多工业和科学问题时高效且准确。尤其是，将机器学习应用于天文学，在分类、聚类和数据清理方面显示出很大的价值。这是因为计算机经过训练后，能够自动执行这些任务，且比人工操作更精确、更迅速。有鉴于此，本文将回顾一些常见的分类算法，并将其中若干种应用于来自斯隆数字巡天的非变星和 RR Lyrae 变星观测数据。为便于比较，我们计算了所用模型的准确率和 F_1-分数。

关键词：机器学习、分类、斯隆数字巡天、非变星、天琴座RR型变星。

1 引言

机器学习是人工智能（AI）和计算机科学的一个分支，专注于利用数据和算法模仿人类的学习方式。机器学习最受欢迎的分支之一是监督学习（Jain，1999）。这是一种机器学习方法，其特点是使用带标签数据集来训练算法，从而对数据进行分类并预测结果。带标签数据集（称为训练集）中的输出标签与输入数据相对应。机器需要理解在未见过的数据中要寻找什么，并基于已从带标签数据中学到的内容给出预测结果。监督式机器学习主要适用于两类问题：分类问题和回归问题。分类是指将输入值映射到一个离散值。在分类问题中，输出通常由类别或类别组成；而回归则与连续数据有关，其预测输出值为实数（Dutton 和 Conroy，1996）。

随着天文学领域的数据规模和复杂性迅速增长，这些变化推动了数据驱动科学的发展，使其成为常见的模型驱动数据分析范式的有益补充。在这一范式中，天文学家开发自动化工具，用于挖掘数据集并从中提取新信息。近年来，机器学习算法在天文学家中越来越受欢迎，机器学习如今已用于各种各样的任务。机器学习算法可应用于天文学和宇宙学领域，执行天体检测和天体分类（Richards et al., 2004）、聚类（Campana et al., 2007）、异常检测（Nun et al., 2014）、特征提取（Sen et al., 2022）以及其他相关任务。

*电子邮箱：zhoolideh@kntu.ac.ir

2 回顾；常用分类算法

2.1 线性回归

简单线性回归是一种回归分析，其中一个自变量（x）与另一个因变量（y）呈线性关系。根据给定的数据点，该算法尝试寻找一条能以最佳方式拟合这些数据点的直线。这条直线可用线性方程建模（Efron & Tibshirani, 1991）

$$y = a * x + b \tag{1}$$

线性回归算法的目标是找到 a 和 b的最佳值。代价函数可以帮助我们找出 a 和 b 的最佳可能值，从而为数据点拟合出最佳直线。由于我们需要找到 a 和 b 的最佳值，因此将这个搜索问题转化为一个最小化问题，即尽可能减小预测值与实际值之间的误差。

$$\text{Minimize} \frac{1}{n} \sum_{i=1}^{n} (y_{pred} - y_i)^2 \tag{2}$$

$$J = \frac{1}{n} \sum_{i=1}^{n} (y_{pred} - y_i)^2 \tag{3}$$

我们选择上述函数进行最小化。预测值与真实值之间的差异用于衡量误差。我们将误差平方后，对所有数据点的平方误差求和，再除以数据点总数，由此得到所有数据点的平均平方误差。现在，利用这个代价函数，我们可以调整 a 和 b的值，使代价函数的值收敛到最小值。要最小化代价函数，可以使用梯度下降法，通过更新 a 和 b 来降低代价函数的值。其基本思路是先为 a 和 b 设定一些初始值，然后迭代调整这些值以降低代价。梯度下降法可以指导我们如何调整这些值。要更新 a 和 b，我们需要求代价函数的梯度。求这些梯度时，我们分别对 a 和 b 求偏导数，具体如下：

$$\begin{aligned}
J &= \frac{1}{n} \sum_{i=1}^{n} (y_{pred} - y_i)^2 \\
J &= \frac{1}{n} \sum_{i=1}^{n} (a + b \cdot x_i - y_i)^2 \\
\frac{\partial J}{\partial a} &= \frac{2}{n} \sum_{i=1}^{n} (a + b \cdot x_i - y_i) \Longrightarrow \frac{\partial J}{\partial a} = \frac{2}{n} \sum_{i=1}^{n} (y_{pred} - y_i) \\
\frac{\partial J}{\partial b} &= \frac{2}{n} \sum_{i=1}^{n} (a + b \cdot x_i - y_i) \cdot x_i \Longrightarrow \frac{\partial J}{\partial b} = \frac{2}{n} \sum_{i=1}^{n} (y_{pred} - y_i) \cdot x_i
\end{aligned} \tag{4}$$

$$\begin{aligned}
a &= a - \alpha \cdot \frac{2}{n} \sum_{i=1}^{n} (y_{pred} - y_i) \\
b &= b - \alpha \cdot \frac{2}{n} \sum_{i=1}^{n} (y_{pred} - y_i) \cdot x_i
\end{aligned} \tag{5}$$

偏导数就是梯度，用于更新 a 和 b的值。 α 是学习率，这是一项必须指定的超参数。较小的学习率有助于更接近极小值，但到达极小值需要更多时间；较大的学习率收敛得更快，但可能会越过极小值。

2.2 逻辑回归

逻辑回归是另一种强大的监督式机器学习算法，用于解决二分类问题（当目标变量为类别型时）。理解逻辑回归的最佳方式是将其视为用于分类问题的线性回归。逻辑回归本质上使用下文定义的逻辑函数来建模二元输出变量（Sur & Cand`es，2019）。线性回归与逻辑回归的主要区别在于，逻辑回归的取值范围限定在 0 到 1 之间。此外，与线性回归不同，逻辑回归不要求输入变量与输出变量之间存在线性关系。在这种方法中，代价函数不是基于 y定义，而是使用 y 的 Sigmoid函数。

$$
\begin{aligned}
&Output = 0 \ or \ 1 \\
&y = a * x + b \\
&\mathrm{h\Theta(x)} = \mathrm{sigmoid(y)} \\
&\mathrm{sigmoid}(y) = \tfrac{1}{1+e^{-y}}
\end{aligned}
\tag{6}
$$

$$
\begin{aligned}
\operatorname{Cost}(h \ominus (x), y(\text{ actual })) = &-\log(h \ominus (x)) \text{ if } y = 1 \\
&-\log(1 - h \ominus (x)) \text{ if } y = 0
\end{aligned}
\tag{7}
$$

根据式（7）定义逻辑回归的代价函数后，只需将该代价函数最小化，就可以轻松求出 a 和 b；因此，y 也将随之确定。

2.3 朴素贝叶斯（NB）

利用贝叶斯定理，我们可以求出在 B 已发生的条件下 A 发生的概率。这里，B 是证据，A 是假设。此处的假设是各预测变量/特征彼此独立。贝叶斯定理可以改写为：

$$
P(y \mid X) = \frac{P(X \mid y) P(y)}{P(X)} \tag{8}
$$

其中，X 是特征向量；将 X 代入并利用链式法则展开后，我们得到：

$$
P\left(y \mid x_1, \ldots, x_n\right) = \frac{P\left(x_1 \mid y\right) P\left(x_2 \mid y\right) \ldots P\left(x_n \mid y\right) P(y)}{P\left(x_1\right) P\left(x_2\right) \ldots P\left(x_n\right)} \tag{9}
$$

现在，可以通过查看数据集来获取每个值，并将其代入方程。对于数据集中的所有条目，分母都保持不变。因此，可以去掉分母，并引入正比关系。

$$
P(y \mid x_1, ..., x_n) \propto P(y) \ \Pi_i^n \ P(x_i \mid y) \tag{10}
$$

一种常见的朴素贝叶斯分类器是高斯朴素贝叶斯。当预测变量取连续值而非离散型值时，我们假设这些值是从高斯分布中采样得到的。由于数据集中的值呈现方式发生了变化，条件概率的公式也随之变为：

$$
P\left(x_i \mid y\right) = \frac{1}{\sqrt{2 \pi \sigma_y^2}} \exp\left(-\frac{\left(x_i - \mu_y\right)^2}{2 \sigma_y^2}\right) \tag{11}
$$

在我们的案例中，类别变量(y)只有两种结果：是或否。分类也可能涉及多个类别。因此，我们需要找出概率最大的类别 y。

$$y = \text{argmax} P(y) \, \Pi_i^n \; P(x_i \mid y) \tag{12}$$

朴素贝叶斯算法主要用于情感分析、垃圾邮件过滤、推荐系统等领域。它们速度快且易于实现，但最大的缺点是要求预测变量彼此独立（du Buisson 等，2015）。在大多数实际案例中，预测变量彼此相关，这会影响分类器的性能。

2.4 支持向量机（SVM）

支持向量机因其以较少的计算能力实现较高的准确率而受到许多人的青睐。支持向量机（简称 SVM）既可用于回归任务，也可用于分类任务，但它广泛应用于分类。支持向量机算法的目标是在 N 维空间中（N——特征的数量）寻找一个能够明确分类数据点的超平面（Vapnik，1995）。为了分离两类数据点，可以选择的超平面有很多。我们的目标是找到一个具有最大间隔的平面，即两类数据点之间距离最大的平面。最大化间隔距离能够提供一定的强化作用，使未来的数据点可以更有信心地被分类。超平面是帮助对数据点进行分类的决策边界。落在超平面任一侧的数据点可以归属于不同的类别。此外，超平面的维度取决于特征的数量。在逻辑回归中，我们取得线性函数的输出，并使用 Sigmoid函数将该值压缩到 [0,1] 的范围内。如果压缩后的值大于阈值（0.5），我们就为其分配标签 1；否则分配标签 0。在 SVM 中，我们取得线性函数的输出；如果该输出大于 1，就将其识别为一个类别；如果输出为 -1，就将其识别为另一个类别。由于 SVM 中的阈值被改为 1 和 -1，我们得到这个由数值（[-1,1]）构成的强化范围，该范围充当间隔。

在支持向量机算法中，我们要最大化数据点与超平面之间的间隔。合页损失有助于最大化间隔。如果预测值与实际值的符号相同，代价为 0；如果不同，则计算损失值。 $c(x, y, f(x)) = \begin{cases} 0 \\ \end{cases}$，

如果 $y * f\ x\ \geq 1$ ( ) ( ) 否则

$1 - y * f\ x$

我们还会在代价函数中添加一个正则化参数。正则化参数的目的是平衡间隔最大化与损失。添加正则化参数后，代价函数如下所示。

$$\min_w [\lambda \|w\|^2 + \sum_{i=1}^{n} (1 - y_i \langle x_i, w \rangle)_+] \tag{13}$$

通过对权重求偏导数来计算梯度。利用梯度，我们可以更新权重。

$$\frac{\delta}{\delta w_k} \lambda \|w\|^2 = 2\lambda w_k \tag{14}$$

$$\frac{\delta}{\delta w_k} (1 - y_i \langle x_i, w \rangle)_+ = \begin{cases} 0, & \text{if } y_i \langle x_i, w \rangle \geq 1 \\ -y_i x_{ik}, & \text{else} \end{cases} \tag{15}$$

当没有误分类时，也就是说，我们的模型正确预测了数据点所属的类别，我们只需根据正则化参数更新梯度。

$$w = w - \alpha \cdot (2\lambda w) \tag{16}$$

发生误分类时，也就是说，模型在预测数据点所属类别时出错，我们会将损失与正则化参数结合起来进行梯度更新。

$$w = w + \alpha \cdot (y_i \cdot x_i - 2\lambda w) \tag{17}$$

2.5 K 均值

K 均值是一种无监督分类算法，也称为聚类，它根据对象的特征将其分为 k 组。分组过程通过最小化每个对象与所属组或聚类质心之间的距离之和来完成（MacQueen，1967）。通常使用的距离是平方距离或欧几里得距离。该算法分为三个步骤：1）初始化：选定组数 k 后，在数据空间中确定 k 个质心，例如随机选择；2）将对象分配给质心：将每个数据对象分配给距离其最近的质心；3）更新质心：更新每组质心的位置，将该组中所有对象位置的平均值作为新的质心。重复步骤 2 和 3，直到质心不再移动，或每一步的移动距离都低于某个阈值。K 均值算法求解一个优化问题，其待优化（最小化）的函数是每个对象到其聚类质心的平方距离之和。对象用 d 维向量表示（$\mathrm{x}_1, \mathrm{x}_2$，...，$\mathrm{x}_n$），K 均值算法构建 k 个组，并使每个组 S $= \{S_1, S_2, \ldots, S_k\}$ 内对象到其质心的距离之和最小化。该问题可以表述为:

$$\min_{\mathbf{S}} E\left(\boldsymbol{\mu_i}\right) = \min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \left\|\mathbf{x}_j - \boldsymbol{\mu}_i\right\|^2 \tag{18}$$

其中，S 是一个数据集，其元素是由向量表示的对象 x_j，其中每个元素都表示一个特征或属性。我们将得到 k 个组或簇及其对应的质心 $\boldsymbol{\mu_i}$。在每次更新质心时，从数学角度来看，我们对函数 $E(\boldsymbol{\mu_i})$施加极值（在本例中为最小值）所需的条件。对于这个二次函数 (1)，该条件为

$$\frac{\partial E}{\partial \boldsymbol{\mu}_i} = 0 \implies \boldsymbol{\mu}_i^{(t+1)} = \frac{1}{\left|S_i^{(t)}\right|} \sum_{\mathbf{x}_j \in S_i^{(t)}} \mathbf{x}_j \tag{19}$$

解法是将每个组中元素的平均值作为新的质心。我们使用了梯度下降法。K 均值方法的主要优点是简单且快速。但我们需要确定 k 的取值，而且最终结果取决于质心的初始化方式。此外，它不一定会收敛到全局最小值，也可能收敛到局部最小值。

2.6 决策树

决策树通过依据特征值对对象进行排序来分类。决策树中的每个节点代表待分类实例的一个特征，每个分支代表该节点可以取的一个值。对象从根节点开始，根据其特征值进行排序和分类。顾名思义，这种方法采用树状的决策模型。在第一次划分（即根节点处），会考虑所有属性/特征，并根据这次划分将训练数据分成若干组（Murthy，1998）。假设我们有三个特征，那么就有三个候选划分。因此，可以使用一个函数计算每种划分的代价，并选择代价最低的划分。当特征数量很大时，划分数量也会随之增加，进而形成一棵庞大的树。这类树结构复杂，可能导致过拟合。因此，我们需要知道何时停止。一种做法是设定模型的最大深度。最大深度指从根节点到叶节点的最长路径长度。

决策树是一种监督式机器学习（也就是说，你需要在训练数据中说明输入是什么，以及相应的输出是什么），数据会根据某个参数不断进行划分。树可以用两个实体来说明，即决策节点和叶节点。叶节点表示决策或最终结果，而决策节点则是数据被划分的位置。目前有许多用于构建决策树的算法，其中一种最优秀的算法称为 ID3 算法。ID3 是迭代二分器3的缩写。在讨论 ID3 算法之前，我们先了解几个定义。

熵，也称为香农熵，对于有限集合 S，记作 $H(S)$，用于衡量数据中的不确定性或随机性。

$$H(S) = \sum_{x \in X} p(x) \log_2 \frac{1}{p(x)} \tag{20}$$

决策树修改版 直观而言，它描述了某个事件的可预测性。例如，考虑一次抛硬币，正面朝上的概率为 0.5，反面朝上的概率也为 0.5。此时熵达到最大值，因为我们无法判断结果会是什么。或者，考虑一枚两面都是正面的硬币。由于我们事先知道结果永远是正面，因此可以完美预测这类事件的熵。换句话说，这个事件没有随机性，因此其熵为零。具体而言，熵值越低，不确定性越小；熵值越高，不确定性越大。

信息增益也称为 Kullback-Leibler 散度，记作 IG(S,A)。对于集合 S，它表示确定某个属性 A 后熵的实际变化。它衡量相对于自变量 $IG(S, A) = H(S) - H(S, A)$ 的熵的相对变化。或者，

$$IG(S, A) = H(S) - \sum_{i=0}^{n} P(x) * H(x) \tag{21}$$

其中， $IG(S, A)$ 表示应用特征 A 所获得的信息增益。 $H(S)$ 表示整个集合的熵，而第二项则计算应用特征 A 后的熵，其中 $P(x)$ 表示事件 x 发生的概率。

ID3 算法将递归执行以下任务。a) 为决策树创建根节点；b) 如果所有示例均为正例，则返回“正例”叶节点；c) 如果所有示例均为负例，则返回“负例”叶节点；d) 计算当前状态的熵 H(S)；e) 对于每个属性，计算相对于属性“x”的熵，记为 H(S, x)；g) 选择 IG(S, x) 值最大的属性；h) 从属性集合中移除提供最高 IG 值的属性；i) 重复此过程，直到所有属性均用尽，或决策树的所有节点均为叶节点。

2.7 神经网络

神经网络是由多层神经元组成的网络，用于对事物进行分类并做出预测。神经网络由密集连接的处理节点组成（Rumelhart 等人，1986），这些节点类似于大脑中的神经元。每个节点都可能与其上方和下方多个层中的不同节点相连接。这些节点以前馈方式在网络中传递数据，即数据只沿一个方向移动。当节点将信息传递给下一个节点时，它会像神经元一样“触发”。一个简单的神经网络包含输入层、输出层，以及位于两者之间的一个隐藏层。包含输入层和输出层在内，层数超过三层的网络称为深度学习网络。在深度学习网络中，每个节点层都会根据前一层的输出对数据进行训练。层数越多，网络基于前面各层的数据识别更复杂信息的能力就越强。网络通过为每个相连节点分配一个称为“权重”的数值来做出决策。权重表示分配给单个节点的信息价值（即该信息对于正确分类信息的帮助程度）。当一个节点接收到来自其他节点的信息时，它会计算该信息的总权重或价值。如果该数值超过某个阈值，信息就会传递到下一层。如果权重低于阈值，信息则不会继续传递。在新构建的神经网络中，所有权重和阈值都被设为随机数。随着训练数据被输入层接收，权重和阈值会不断调整，从而持续产生正确的输出。在图（1）中，可以看到神经网络的不同部分。

3 数据

这里我们将介绍一组真实的天文数据样本。天琴座RR型变星是周期变星，常见于球状星团中（Ivezic 等，2005年）。它们被用作标准

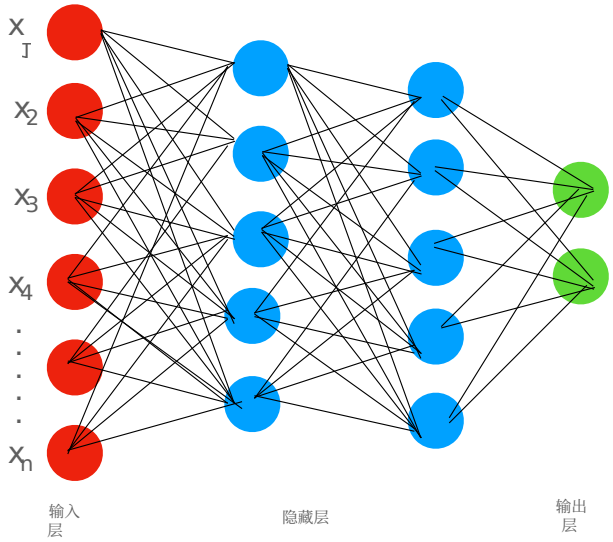

图 1：神经网络的示意图。如图所示，它主要由三个部分组成：输入层、隐藏层和输出层。

烛光来测量星系距离，助力构建宇宙距离阶梯。它们是A或F型光谱的脉动水平分支星，质量约为太阳的一半。据认为，它们在红巨星分支阶段失去了部分质量；它们曾是质量与太阳相近或略小的恒星，约为0.8个太阳质量。在当代天文学中，周期-光度关系使它们成为测量相对近距离目标的优良标准烛光，尤其适用于银河系和本星系群内的目标。它们也是球状星团研究中的常见对象。我们使用斯隆数字巡天中对天琴座RR型变星进行的测光观测作为数据。该数据集来自斯隆数字巡天条带82，并结合了条带82标准星（代表对非变星的观测）以及从与标准星相同的观测中提取、并依据补充数据按其变异性筛选出的天琴座RR型变星。根据$0.7 < u - g < 1.35$, $-0.15 < g - r < 0.4, -0.15 < r - i < 0.22, and -0.21 < i - z < 0.25$(Ivezic 等，2005年)，样本还进一步限定在整体颜色-颜色空间中一个较小的区域内。这些筛选标准最终得到包含92,658颗非变星和483颗天琴座RR型变星的样本。这个组合数据集的两个特征使其非常适合用于测试分类算法：

4 分类指标

进行分类预测时，可能出现四种结果。真正例是指预测某个观测值属于某个类别，而它实际上也确实属于该类别。真负例是指预测某个观测值不属于某个类别，而它实际上也确实不属于该类别。假正例是指预测某个观测值属于某个类别，但实际上并非如此。假负例是指预测某个观测值不属于某个类别，但实际上它确实属于该类别。这四种结果通常会绘制在混淆矩阵中。对测试数据进行预测后，可通过将每项预测归为上述四种可能结果之一来生成该矩阵。

用于评估分类模型的四项主要指标是准确率、精确率、召回率和 F_1 分数。

分类器	准确率	F_1
逻辑回归	0.969	0.628
朴素贝叶斯	0.983	0.654
SVMs	0.994	0.499
决策树	0.994	0.703
神经网络	0.993	0.512

表 1：六种选定分类器的准确率和 F_1 得分。虽然所有模型都表现出很高的准确率，但不同模型的 F_1 有所不同，因此它是评估模型的更好指标。

准确率是测试数据中预测正确的比例。计算方法很简单：用预测正确的数量除以预测总数。

$$\text{accuracy} = \frac{\text{correct predictions}}{\text{all predictions}} \tag{22}$$

精确率定义为相关样本（真正例）占所有被预测为属于某一类别的样本的比例。

$$\text{precision} = \frac{\text{true positives}}{\text{true positives+false positives}} \tag{23}$$

召回率定义为被预测为属于某一类别的样本数占所有实际属于该类别的样本数的比例。

$$\text{recall} = \frac{\text{true positives}}{\text{true positives+false negatives}} \tag{24}$$

F_1得分通过计算分类器精确率和召回率的调和平均数，将二者合并为一个指标。它主要用于比较两个分类器的性能。假设分类器 A 的召回率较高，而分类器 B 的精确率较高。在这种情况下，可以利用这两个分类器的 $F1$得分来判断哪一个能产生更好的结果。

$$F_1 = \frac{2 \text{ precision } * \text{ recall}}{\text{precision } + \text{ recall}} \tag{25}$$

5 结果

在本节中，我们将多种常用分类器应用于数据集，训练一系列机器学习模型，并对这些模型进行比较。图（2、3）展示了样本数据，以及逻辑回归和高斯朴素贝叶斯分类器的分类边界。表（1）报告了不同模型的准确率和 F_1-分数，可用于比较所提出的模型。

6 总结与结论

我们回顾了一些热门的分类算法，包括线性回归、逻辑回归、朴素贝叶斯、支持向量机、决策树和神经网络。随后，我们将一些提出的模型应用于从斯隆数字化巡天 Stripe 82 中提取的变星和非变星数据集。在该数据集上训练模型后，我们对模型进行了测试，并评估了预测结果。我们使用准确率和 F_1-分数作为指标，结果见论文表（1）。如表所示，决策树算法的准确率和F_1-分数最佳。此外，为了便于理解，我们在图（2、3）中绘制了逻辑回归模型和朴素贝叶斯模型的决策边界。

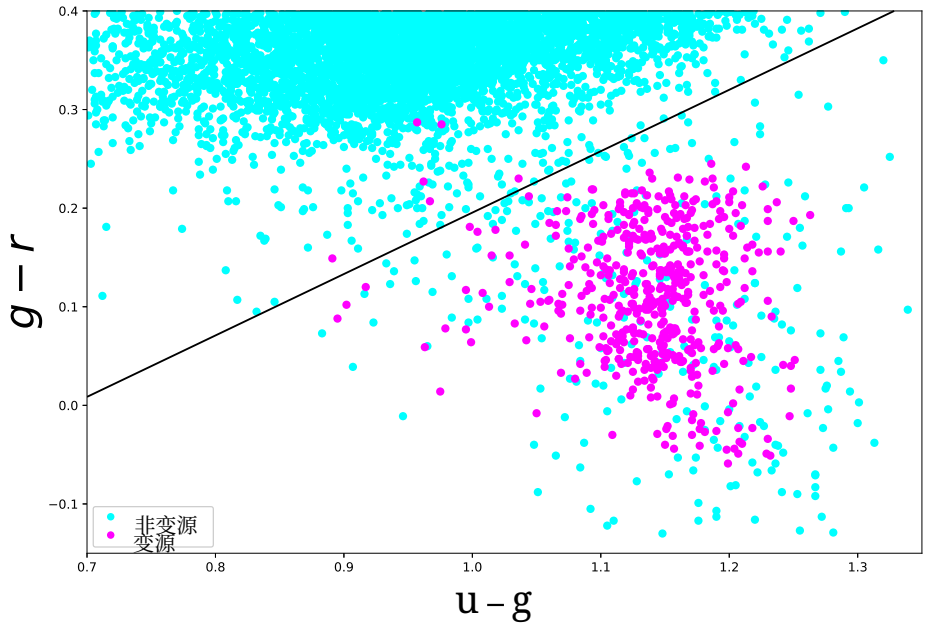

图 2：采用逻辑回归分类方法将变星与非变星分开。浅蓝色点表示非变源，紫色点表示变源。黑线表示分类边界。逻辑回归的准确率为 0.969， F_1-分数为 0.628。

致谢

作者感谢 ICRANet – 伊斯法罕天文学会议的组织者诚挚邀请其为本次活动作报告。

参考文献

Campana R.、Massaro E.、Gasparrini D.、Cutini S.、Tramacere A.，2007，AIPC，921，536。doi:10.1063/1.2757439

Dutton D、Conroy G（1996）机器学习综述。《Knowl Eng Rev》12：341–367

Efron B.、Tibshirani R.，1991，Sci，253，390。doi:10.1126/science.253.5018.390

Jain AK、Murty MN、Flynn P（1999）数据聚类：综述。《ACM Comput Surveys》31(3)：264–323

MacQueen, J.B.（1967），《伯克利数学统计与概率研讨会》第5届会议论文集，第1卷：统计学，加利福尼亚大学出版社，伯克利，281-297。

Murthy，（1998），多学科综述，《数据挖掘与知识发现》2：345–38 9.

Ivezi c Z.、Vivas A. K.、Lupton R. H.、Zinn R.（2005年）。利用 g si单历元数据筛选RR Lyrae变星。《天文学杂志》129，1096–1108。

Nun I., Pichara K., Protopapas P., Kim D.-W., 2014年，《天体物理学杂志》，793，23。doi:10.1088/0004- 637X/793/1/23

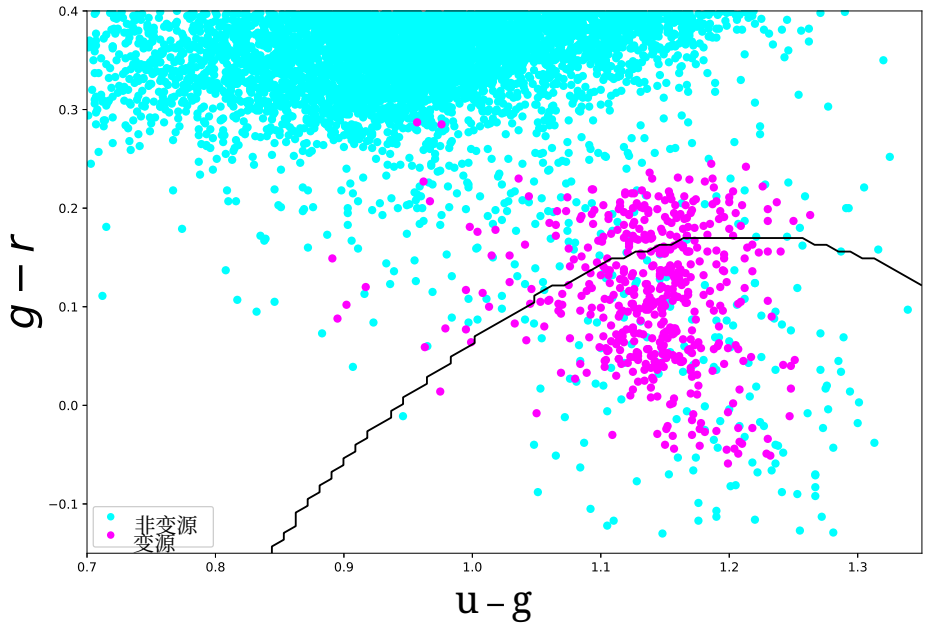

图 3：使用高斯朴素贝叶斯分类方法将变星与非变星分离。浅蓝色点表示非变源，紫罗兰色点表示变源。黑线表示分类边界。朴素贝叶斯的准确率达到 0.983， F_1-分数为 0.655。

du Buisson L., Sivanandam N., Bassett B. A., Smith M., 2015，皇家天文学会月刊，454，2026。doi:10.1093/mnras/stv2041 Richards G. T., Nichol R. C., Gray A. G., Brunner R. J., Lupton R. H., Vanden Berk D. E., Chong S. S., 等，2004，天体物理学杂志增刊系列，155，257。doi:10.1086/425356 Rumelhart, D. E., Hinton, G. E., Williams, R. J.（1986），《并行分布式处理：认知的微观结构探索》。麻省理工学院出版社，马萨诸塞州剑桥，第1卷，第318-362页。Sen S., Agarwal S., Chakraborty P., Singh K. P., 2022，实验天文学，53，1。doi:10.1007/s10686-021-09827-4 Sur P., Cand`es E. J., 2019，美国国家科学院院刊，116，14516。doi:10.1073/pnas.1810420116 Vapnik, V.，1995，《统计学习理论的本质》。施普林格出版社。