决策树 ID3 算法在 Iris 数据集的应用

吕书礼 **18375305**

摘要

决策树作为机器学习中常见分类算法中的一种，生成算法有 ID3, C4.5 和 C5.0 等。J. Ross Quinlan 在 1975 年提出了 ID3 算法掀起了决策树研究的高潮，并在 1993 年提出的 C4.5 算法。决策树的生成过程，就是使用满足特征选择准则的特征不断的将数据集划分为纯度更高，不确定性更小的子集的过程。

本文使用 ID3 算法对 Iris 数据集进行划分，实现了 ID3 算法分割 Iris 数据集的底层代码实现并进行了决策树可视化，同时与 sklearn 库中拟合得到的决策树结果进行了对比。为了提升模型的泛化能力和鲁棒性，采用剪枝算法增大算法自由度，拟合选取合适参数进行剪枝。使用 sklearn 库实现了高斯朴素贝叶斯算法在改变数据集大小条件下的实现效果，得出高斯朴素贝叶斯算法的特点并与决策树特点的相比较。最后对软决策树进行了简单的实验，并通过对软决策树相关文献的研究，发现决策树与深度神经网络的联系，对二者的发展方向进行了探究与学习。

关键词：决策树，Iris 数据集，剪枝，朴素贝叶斯算法，软决策树

一．算法介绍与基础

决策树是一种树形结构，其中每个内部节点表示一个属性上的判断，每个分支代表一个判断结果的输出，最后每个叶节点代表一种分类结果。

ID3 算法即 Iterative Dichotomiser 3，最早是由昆兰（J. Ross Quinlan）于 1975 年提出的一种分类预测算法[1]，算法以信息论为基础，以信息熵和信息增益度为衡量标准，从而实现对数据的归纳分类。

信息熵表现出信息的不确定度：

$$Entropy(n) = -\sum_{i=0}^{N-1} p(i \mid n) \log_2 p(i \mid n) \tag{1}$$

其中，n 代表节点 n，i 代表分类数，$p(i \mid n)$ 表示在节点 n 分类为 i 的概率。

信息增益指的就是划分可以带来纯度的提高，信息熵的下降。它的计算公式，是父节点的信息熵减去所有子节点的信息熵。在计算的过程中，我们会计算每个子节点的归一化信息熵，即按照每个子节点在父节点中出现的概率，来计算这些子节点的信息熵。所以信息增益的公式可以表示为：

$$G(F,a) = Entropy(F) - \sum_{i=1}^{k} \frac{|s_i|}{|F|} Entropy(s_i) \tag{2}$$

其中，F 代表父节点，a 为划分使用特征，s_i 为子节点。

ID3 算法通过计算每个属性的信息增益，认为信息增益高的是优属性，每次划分选取信息增益最高的属性为划分标准，对数据集进行划分，重复这个过程，直至生成一个能完美分类训练样例的决策树。

二．Iris 数据集

Iris 数据集[2]是常用的分类实验数据集，由 Fisher, 1936 收集整理。Iris 也称鸢尾花卉数据集，是一类多重变量分析的数据集。数据集包含 150 个数据样本，分为 3 类，每类 50 个数据，每个数据包含 4 个属性。可通过花萼长度(Sepal.Length)，花萼宽度(Sepal.Width)，花瓣长度(Petal.Length)，花瓣宽度

(Petal.Width)4 个属性预测鸢尾花卉属于(Setosa，Versicolour，Virginica)三个种类中的哪一类。

将特征两两组合成二维子空间，将数据分别映射至子空间，并可视化数据集如下所示:

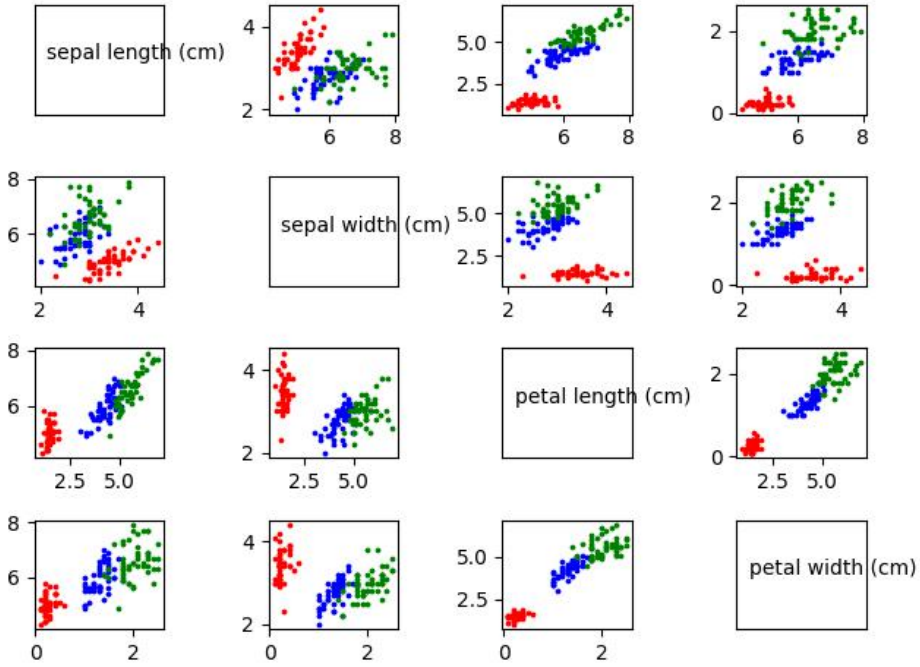

图 1. Iris 数据集可视化结果

可以直观看出，红色点与蓝绿色点线性可分，蓝绿色点存在交叉，不能直接线性分解。因此很难使用线性分类器直接完全分类，且存在一类可以线性分割。

三．算法流程

本次实验算法分为数据集处理，连续特征值离散化处理，通过信息增益进行特征与特征划分点选取，递归迭代生成树，预剪枝算法，预剪枝参数拟合，混淆矩阵测试等几个部分。

算法流程图如下所示：

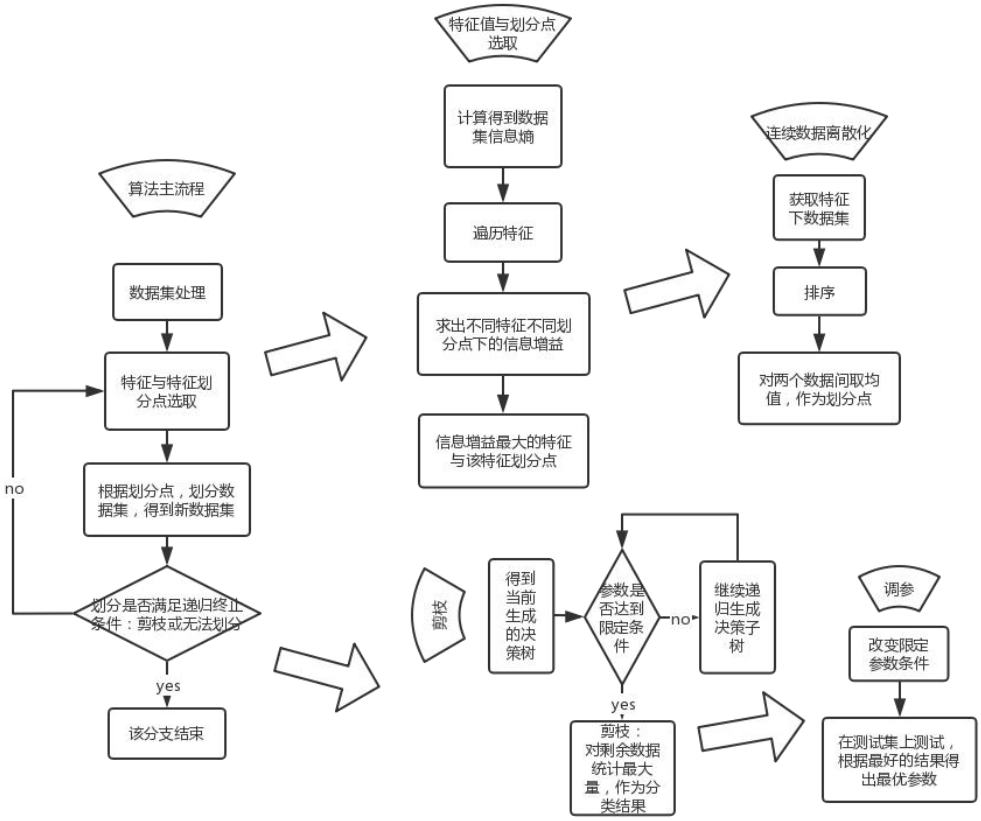

图 2. 算法流程图

3.1 数据集处理

3.1.1 数据集混洗与划分

首先将 Iris 数据集进行随机打乱，然后以 0.6：0.2：0.2 的比例生成数据集，交叉验证集与测试集。其中，交叉验证集用于选取剪枝参数。将处理完备的测试集以静态文件的形式保存，方便测试使用。

3.1.2 连续数据离散化处理

使用二分法进行离散化处理。对于给定样本集$\{x_1, x_2, x_3, \cdots, x_n\}$，首先进行排序处理，对排序结果的点集，两相邻点之间取平均值$t_i = \frac{x_i + x_{i+1}}{2}$作为划分点，构成划分点集$T$。对任意一个划分点，即可使用离散属性决策树的方法进行考察。

3.2 特征与划分点的选取

对每次数据集划分，都需要对传入数据的进行划分特征的选取和相应特征划分点的选取，选取方式为判断任意划分结果的信息增益，选择信息增益最大的划

分选取方式作为传入数据集的划分方式，也即选取特征和选取特征对应的划分点。

1. 计算传入数据的信息熵。
2. 得到不同特征与其对应的不同划分点(连续数据离散化处理)。
3. 计算所有特征与其所有划分点的信息增益。
4. 返回最优特征值及其划分点。

3.3 剪枝

剪枝(pruning)的目的是为了避免决策树模型的过拟合。因为决策树算法在学习的过程中为了尽可能的正确的分类训练样本，不停地对结点进行划分，因此这会导致整棵树的分支过多，也就导致了过拟合。决策树的剪枝策略最基本的有两种：预剪枝(pre-pruning)和后剪枝(post-pruning)。

预剪枝就是在构造决策树的过程中，先对每个结点在划分前进行估计，如果当前结点的划分不能带来决策树模型泛化性能的提升，则不对当前结点进行划分并且将当前结点标记为叶结点。

后剪枝就是先把整颗决策树构造完毕，然后自底向上的对非叶结点进行考察，若将该结点对应的子树换为叶结点能够带来泛化性能的提升，则把该子树替换为叶结点。

本文参考 sklearn 库的 tree.DecisionTreeClassifier()函数中设定参数用于限制最大层数和最大样本量的思想，进行预剪枝算法编写，在底层代码中加入该剪枝环节，对于满足剪枝条件的分类，统计经过该节点的样本标签，选取占比较高的作为分类结果。

对于参数的选取，为了避免在测试集上选取是对测试集的拟合过程，引入交叉验证集，本文采用在比较在交叉验证集下不同参数下分类最优结果的方式，选取参数。经检验，有效的增加了模型的鲁棒性和泛化能力。

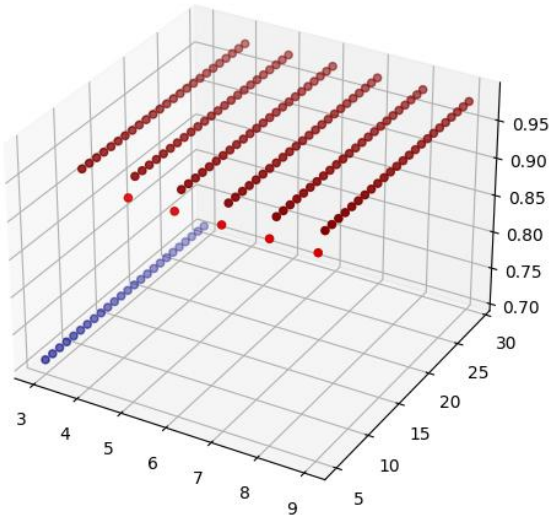

图 3. 剪枝参数在交叉验证集上的表现

如图为参数改变下准确率的变化散点图，x 轴为最大深度参数，y 轴为停止划分数据量参数，颜色相同的具有相同的准确率，由此可以看出，点靠图左后方对应的模型复杂度最低，因此，在相同准确率的情况下，选择左后方的剪枝参数，可以较好的避免过拟合与增强模型的鲁棒性和泛化能力。

3.4 递归创建决策树

本文选取的决策树数据结构为二叉树，存储方式为字典。生成决策树的算法流程如下，所示：

1.判断递归是否满足剪枝条件或终止条件，满足则停止递归。

2.判断数据集是否只含有一类标签，若含有则递归结束。

3.对数据集离散化处理。

4.根据划分点集进行特征与划分点选取。

5.根据划分结果划分数据集。

6.将划分的数据集递归重复上述步骤。

3.5 测试：

构建混淆矩阵进行测试，同时对剪枝前后准确率进行了对比。

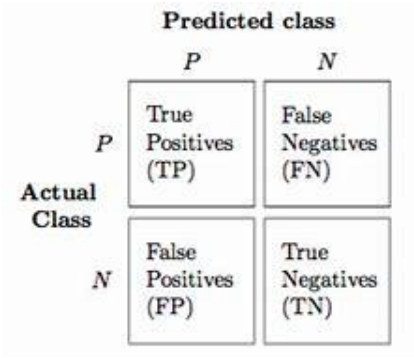

图 4. 混淆矩阵图例

对比所用指标选取如下：

$$
\begin{aligned}
precision &= \frac{TP}{TP+FP} \\
recall &= \frac{TP}{TP+FN} \\
accuracy &= \frac{TP+TN}{TP+TN+FP+FN} \\
F1\_score &= \frac{2}{\frac{1}{recall}+\frac{1}{precision}} = \frac{2TP}{2TP+FP+FN}
\end{aligned}
\tag{3}
$$

四．结果与分析

实现的 ID3 底层算法[3]，生成决策树的结果可视化[4]如下所示：

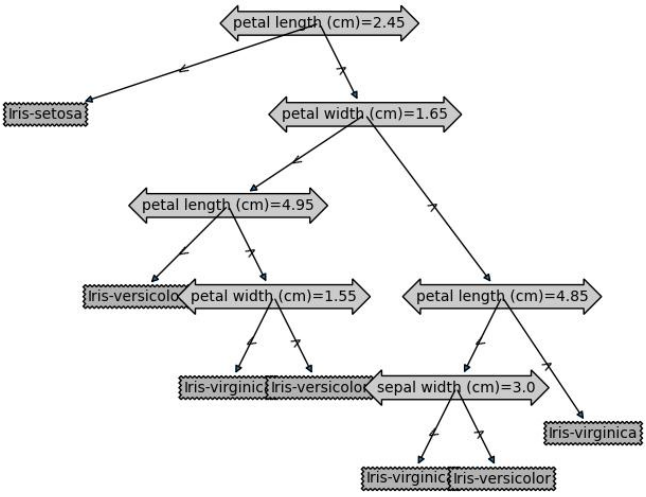

图 5. 底层 ID3 算法实现决策树

对决策树进行测试，测试结果具有较好的准确性，因此可以判断未发生过拟合或轻微过拟合，推断，由于 Iris 数据集本身不大，加之特征少且只进行线性分割拟合也具有很好的效果，所以决策树不会产生较为严重的过拟合：

	precision	recall	F1_score
Iris-setosa	1.0	1.0	1.0
Iris-versicolor	0.92	0.92	0.92
Iris-virginica	0.94	0.94	0.94
aaccuracy = 0.96			

图 6. 底层测试结果

使用 sklearn 构建出的决策树，由于训练集和测试集的随机分割，所以生成模型具有一定的变化，图片选取底层实现和 sklearn 生成决策树进行对比。

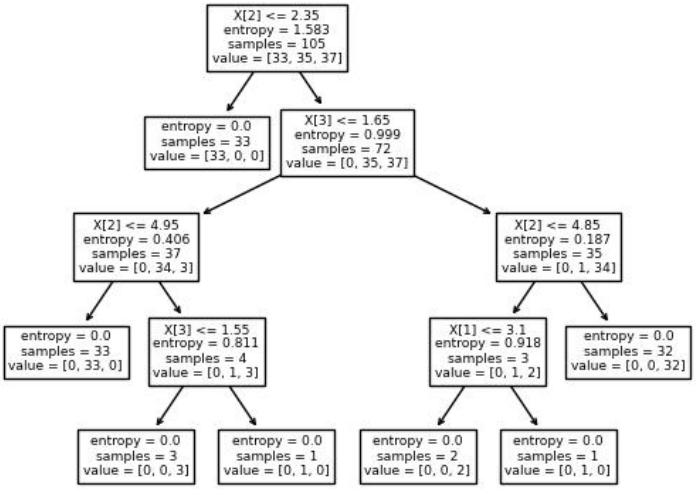

图 7. sklearn 生成决策树图例

拟合结果在测试集上准确率也为 95%以上。

剪枝前后树的模型复杂度对比：

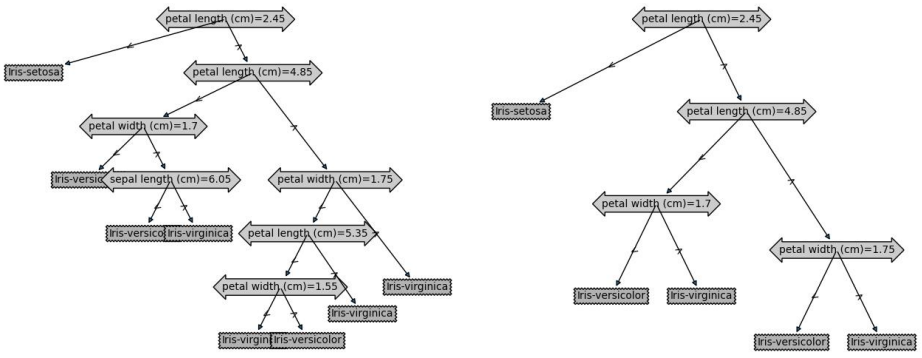

图 8. 剪枝可视化对比模型复杂度

```
                precision   recall      F1_score
Iris-setosa     1.0     1.0     1.0
Iris-versicolor     0.83    0.83    0.83
Iris-virginica  0.89    0.89    0.89
accuracy = 0.91
after pruning
                precision   recall      F1_score
Iris-setosa     1.0     1.0     1.0
Iris-versicolor     0.86    0.83    0.84
Iris-virginica  1.0     0.89    0.94
accuracy = 0.96
```

图 9. 计算结果对比泛化能力

分析结果，进行剪枝后，模型的复杂度肉眼可见的降低了(树的分叉变少了)，对应在测试集上，经过数十次验证，进行剪枝后的模型正确率总是优于未剪枝情况，所以，模型的泛化能力和鲁棒性也未因为剪枝产生影响，反而表现得更加出色了。

五．算法对比

5.1 朴素贝叶斯算法

最为广泛的两种分类模型是决策树模型和朴素贝叶斯模型。和决策树模型相比，朴素贝叶斯分类器(Naive Bayes Classifier 或 NBC)发源于古典数学理论，有着坚实的数学基础，以及稳定的分类效率。同时，NBC 模型所需估计的参数很少，对缺失数据不太敏感，算法也比较简单。理论上，NBC 模型与其他分类方法相比具有最小的误差率。但是实际上因为 NBC 模型假设属性之间相互独立，这个假设在实际中往往是不成立的，这给 NBC 模型的正确分类带来了一定影响。

朴素贝叶斯算法核心公式如下：

$$\hat{y} = \arg\max(y)\prod_{i=1}^{N} P(x_i \mid y) \tag{4}$$

采用 sklearn 中库函数进行测试，使用高斯朴素贝叶斯方法，即假定对任意分类下的所有特征满足高斯分布：

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(x_i-\mu_i)^2}{2\sigma_i^2}} \tag{5}$$

其中，y 表示分类，x_i 表示特征。μ_i, σ_i 表示该分类下特征的均值和标准差。

为验证朴素贝叶斯对少量数据同样适用，改变训练数据集占总数据集的比例，验证准确率。如图所示得到模型准确率与训练数据量 20%到 80%的关系图，左侧为 20%右侧为 80%。由此可以看出朴素贝叶斯算法对数据量的要求较小，在小数据量的情况下仍然保持较好的分类特性。

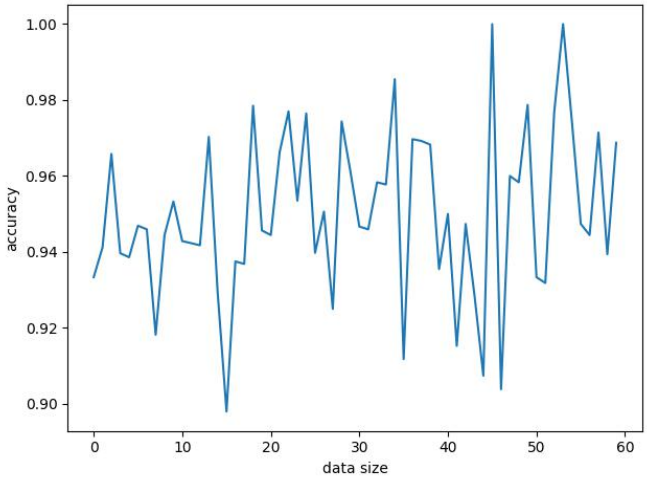

图 10. 计算结果对比泛化能力

对比决策树，在算法思想方面，决策树与朴素贝叶斯算法分别从熵和概率两个完全不同的角度来对数据集进行分类。

决策树具有以下优点：

1）创建决策规则的过程简单。

2）决策树可以可视化，输出结果易于理解。

3）可以处理连续性和离散性数据。

同时具有容易产生过拟合的缺点，可以通过剪枝来缓解。

ID3 的算法规则相对简单，可解释性强。同样也存在缺陷，比如我们会发现 ID3 算法倾向于选择取值比较多的属性。因此可能会选择对分类无效果的属性选择为最优属性。因此 C4.5 算法引入信息增益比以解决此类问题。

5.2 软决策树

相较于从训练数据中直接学习的硬边界决策树，软决策树的泛化能力更强。2003 年，Olaru[5]等人提出了一种特殊的模糊决策树：软决策树。这个决策树具有优越的性能，因此受到关注。2016 年到 2017 年间，研究学者们开始将软决策树于神经网络相结合，充分展示了软决策树的优越效果。例如，2017 年 Hinton

提出了将深度神经网络对应到软决策树上，对神经网进行了一个形象的且合理的解释。

深度网络通过对训练数据的输入和输出之间关系中的大量弱统计规律进行建模从而做出可靠的决策。相比之下，决策树是如何进行任意特定的分类就很容易解释。

以 Distilling a Neural Network Into a Soft Decision Tree[6]一文为例，该文章中使用：

$$p_i(x) = Sigmoid(xw_i + b_i) \tag{6}$$

作为内部决策节点的决策条件，以二叉树的形式进行决策分割，相比于硬决策树，该决策单元引入了非线性环节和可调参数，并采用梯度下降学习参数，所以具有更好的泛化能力和拟合能力。

本文对软决策树进行了一定的尝试，在测试过程中通过概率对决策进行简单的划分，如以 0.8 的概率流向一边，以 0.2 的概率流向另一边。经过测试，由于概率分布难以预测且参数难以估计，所以，对于 Iris 数据集，难以检验软决策数真正的效果。

```
                precision   recall      F1_score
Iris-setosa     0.69    1.0     0.82
Iris-versicolor     0.89    0.73    0.8
Iris-virginica  1.0     0.62    0.77
accuracy = 0.8
after pruning
                precision   recall      F1_score
Iris-setosa     0.92    1.0     0.96
Iris-versicolor     0.92    0.73    0.81
Iris-virginica  1.0     0.62    0.77
accuracy = 0.9333
```

图 11.简单软决策树测试结果

但对于深度神经网络决策树，可以将决策树在每一次决策进行十分准确的分割（对应理解为神经网络一对多的分类问题），即可以将一类结果准确得出，从而在之后的决策中不再考虑此类。对应 Iris 数据集即进行两次决策，即可得到结果。由此可以看出，深度神经网络决策树的决策次数可以预先设定，即对树的结构可以预先设定，但训练过程可能需要较多的数据量。

软决策树有几个优点[7]：首先，它们提供了软分割，而硬决策树在叶子边界上有不连续性。这使得软树能够有更平稳的拟合，从而降低了在分割边界周围的偏差。其次，线性门控函数使软树能够进行倾斜分割，这与硬树的轴正交分割相反，减少了解决回归或分类问题所需的节点数量。准确率方面，软树表现更优。

同时使用决策树可以对深度神经网络的解释性有更好的提升。因此软决策树对与深度神经网络和决策树领域都有着比较重要的影响。

六．结论与思考

对于一个未经过剪枝的决策树，剪枝算法可以有效增加泛化能力，降低模型复杂度。根据本文提出的思路，可以通过在交叉验证集上对增设自由度参数进行调整测试，得到最优剪枝参数，即可找到模型紧凑型与性能的最优策略。同时基于本文自由度，可以增设自由度如特征被选取次数等参数进行拓展，以获得更好的剪枝策略。

对比经典决策树与朴素贝叶斯算法，两算法均具有较好的直观意义上的可解释性。两算法对数据量的要求均不高，朴素贝叶斯尤甚。决策树的弊端在于容易产生过拟合，朴素贝叶斯的弊端在于其假设中独立性往往难以满足。

对于经典决策树，剪枝算法和软决策树方法均能够增加其模型泛化能力。软决策树算法将非线性环节引入决策树模型，实现了决策的平滑处理，并在非线性环节中引入可调参数，可以对参数进行梯度下降学习，即实现了前向传播和反向传播的结合。将决策树与深度神经网络进行联合解释，不仅对于决策树的发展有帮助，同时为解释深度神经网络，提供了思路与方向。

参考文献

[1] J.Ross Quinlan: Predicting The Length Of Solutions To Problems. IJCAL 1975:363-369

[2] UCI Machine Learning Repository: Iris Data Set

[3] https://zhuanlan.zhihu.com/p/136437598

[4] https://www.cnblogs.com/further-further-further/p/9429257.html

[5] Cristina Olaru, Louis Wehenkel, Erratum to “A complete fuzzy decision tree technique”

[6] Nicholas Frosst, Geoffrey E. Hinton: Distilling a Neural Network Into a Soft Decision Tree. CEx@AI*IA 2017

[7]O. İrsoy, O. T. Yıldız and E. Alpaydın, "Soft decision trees," Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), 2012, pp. 1819-1822.