决策树

介绍

人工智能已经发展了多种技术来建造分类模型，其中决策树是一种常用的模型。决策树曾在相当长的时间内是一种非常流行的人工智能技术。20世纪80年代，它是构建人工智能系统的主要方法之一。20世纪90年代初，这一技术随着人工智能遭遇低潮而逐渐不为人所注意。然而，20世纪90年代后期，随着数据挖掘技术的兴起，决策树作为一个构建决策系统的强有力的技术而重新浮出水面。随着数据挖掘在商业智能等方面的应用，决策树技术将在未来发挥越来越强大的作用。

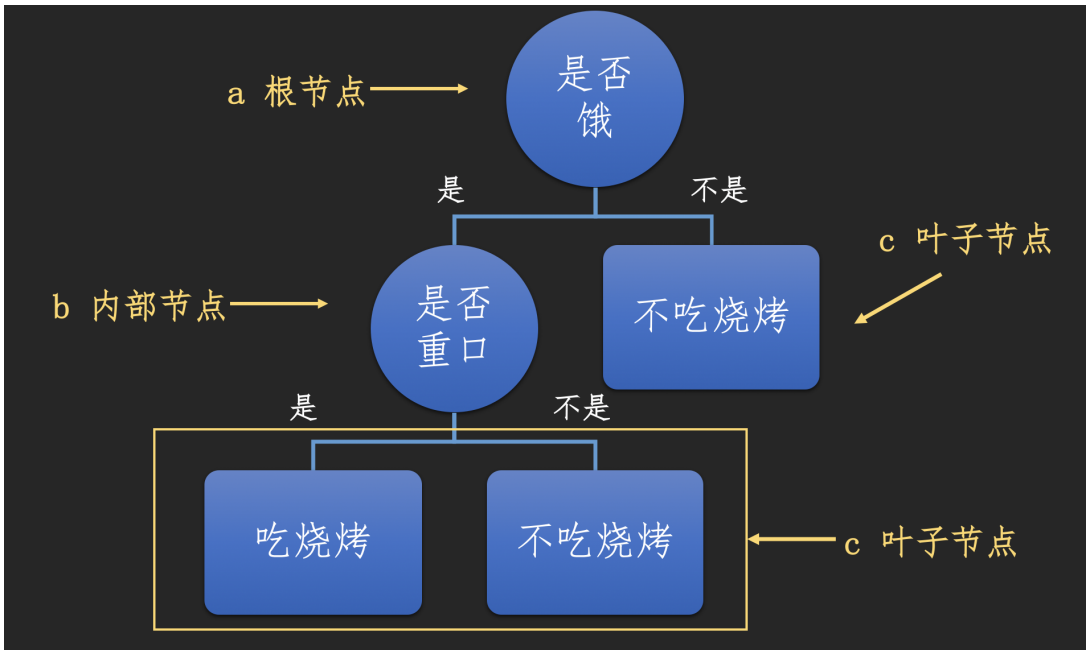
决策树是一种逻辑简单的机器学习算法，采用树形结构，在分类问题中，使用层层推理来实现最终的分类。一般，一棵决策树包含一个根节点，若干个内部节点（非叶子节点）和若干个叶子节点，如下图的a,b,c节点。

根节点：第一个选择点（对决策结果影响最重要的特征）

内部节点（非叶子节点）：中间决策过程

叶子节点：最终的决策结果

以”老赵吃不吃烧烤”为例讲述构建决策树过程。



如上图所示，“是否饿”是根节点；“是否重口”是内部节点，中间决策过程；“吃烧烤”和“不吃烧烤”是叶子节点，最终的决策结果。因此，如何定义和找到最优的特征（“是否饿”，“是否重口”），并找到最好的划分（对于“是否饿”这个特征，“是”和“不是”是最好的划分；同理对于“是否重口”这个特

征，“是”和“不是”也是最好的划分；），是实现决策过程（老赵是否吃烧烤）的主要工作。

那么什么是特征呢?

特征通常指对结果影响比较大的变量。举个例子：机器学习的任务是预测老赵夜宵吃不吃烧烤，如果老赵不饿，他就不吃烧烤，如果老赵饿了，那么他很有可能去吃烧烤。对于老赵夜宵吃不吃烧烤这个任务，是否饿就是一个特征。

决策树分类包括两个步骤：第一步是利用训练样本集来建立并精化出一颗决策树，建立决策树模型。这个过程实际上是一个从数据中获取知识，进行机器学习的过程。通常分为两个阶段：建树和剪枝。第二步是利用建好的决策树对新的数据进行分类。

决策树通过不断选择最优特征划分数据集，对划分后的子数据集不断迭代，从而选择最优特征划分，直到所有的数据集属于同一类别，或者没有特征可以选择为止。选择最优特征的算法有很多种，今天就给大家讲一种最经典的ID3（Iterative Dichotomozer 3）决策树算法，其用信息增益选择最优特征。

决策树的主要优点

生成一颗决策树是从数据中生成分类模型的一个非常有效的方法，相对于其它分类方法，决策树算法应用最为广泛，其独特的优点包括：

- 学习过程中使用者不需要了解很多背景知识，只要训练事例能够用特征--结论的方式表达出来，就能用该算法进行学习；
- 与神经网络等分类算法相比，决策树的训练时间相对较少;
- 决策树的分类模型是树状结构，简单直观，比较符合人类的理解方式；
- 可以将决策树中到达每个叶节点的路径转为if – then形式的分类规则，这种形式更有利于理解。

ID3决策树

在ID3决策树算法的学习过程中，信息增益是特征选择的一个重要指标，它定义为一个特征能够为分类系统带来多少信息，带来的信息越多，说明该特征越重要，相应的信息增益也就越大。也就是说，ID3决策树有多个特征，分别算出每个特征的信息增益，选取信息增益最大的特征。

信息增益的公式为：信息增益 = 信息熵 – 条件熵。那信息熵，条件熵是什么呢?

在构建ID3决策树之前，先来了解几个概念！

先来理解一下什么是信息，信息是一个很抽象的概念，泛指人类社会传播的一切内容。

那信息可以被量化嘛？

当然可以！香农提出的“信息熵”解决了这一问题。

熵 (entropy) 这一词最初来源于热力学。1948年，克劳德·爱尔伍德·香农将热力学中的熵引入信息论，所以也被称为香农熵 (Shannon entropy)，信息熵 (information entropy)。

香农提出：一条信息的信息量大小和它的不确定性有直接的关系。

比如说，我们要搞清楚一件非常非常不确定的事，就需要了解“大量的”信息才能对这件事有更强的确定性。相反，如果我们对某件事已经有了较多的了解，我们只需要“小量的”信息就能把它搞清楚。

举个例子：北京开课吧的员工想知道：北京明天会不会下雪？我们没法立即确定这件事情的答案。为了搞清楚，我们可以看天气预报，这需要额外的信息，说明“北京明天会不会下雪”这件事不确定性大，信息熵高。再比如：北京昨天下没下雪，是确定性事件，信息熵很低。

根据香农给出的信息熵公式，如果一个系统存在多个事件{x1,x2,...,xn}，每个事件的发生概率为{p1,p2,...,pn}，则整个系统的熵为：

$$H(X) = -\sum_{x \in X} p(x) \log p(x)$$

举个例子：一朵百合开花或不开花是一个随机事件，用随机变量X表示。现有一些样本X={开，开，开，不开，不开}。

每个事件发生的概率有：p(X=‘开’) = 3/5；p(X=‘不开’)=2/5，那么X的熵H(X)=–(3/5)log(3/5)–(2/5)log(2/5)

除此之外，我们还需要了解一个条件熵！

条件熵H(Y|X)表示在已知随机变量X的条件下随机变量Y的不确定性。

$$H(Y|X) = \sum_{x \in X} p(x) H(Y|X = x)$$

依旧利用百合花的例子了解条件熵。

假设现在使用百合是否开花来预测天气是晴朗的还是有雨的。所以引入天气变量(Y)：晴、雨。晴天的概率表示为：p(晴)=3/5，雨天的概率表示为：p(雨)=2/5。

晴：开、开、不开	雨：开、不开

现在我们就有了在百合是否开花（X）的条件，天气是晴朗的还是有雨的（Y）概率分布。所以H(Y|X) = 3/5[-(2/3)log(2/3)-(1/3)log(1/3)]+2/5[-(1/2)log(1/2)-(1/2)log(1/2)]

ID3算法采用分治策略，在决策树各级结点上选择属性时，用信息增益作为属性的选择标准，以便在每一个非叶结点上进行测试时，能获得关于被测试记录最大的类别信息。具体方法是：检测所有的属性，选择信息增益最大的属性产生决策树结点，由该属性的不同取值建立分支，再对各分支的子集递归调用该方法建立决策树结点的分支，直到所有子集仅包含同一类别的数据为止。最后得到一颗决策树，它可以对新的样本进行分类。

ID3算法的优点是：算法的理论清晰，方法简单，学习能力较强，分类速度快，适合于大规模数据的处理。主要缺点有：ID3算法只能处理离散性的属性；

信息增益度量存在一个内在偏置，在计算时会偏袒具有较多取值的属性，但有时取值较多的属性不一定是最优的。ID3算法是非递增学习算法，抗噪性能差，训练例子中正例和反例较难控制。

如何对决策树进行剪枝?

一颗完全生长的决策树所对应的每个叶节点中只会包含一个样本，那么这样就会面临一个很严重的问题，即过拟合。用该决策树进行预测时，在测试集上的效果将会很差。因此需要对决策树进行剪枝，剪掉一些树枝，提升模型的泛化能力。

决策树的剪枝通常有两种方法，预剪枝（pre-pruning）和后剪枝（post-pruning）。预剪枝，即在生成决策树的过程中提前停止树的增长。而后剪枝，是在已生成的过拟合决策树上进行剪枝，得到简化版的剪枝决策树。

预剪枝的核心思想是在树中结点进行扩展之前，先计算当前的划分是否能带来模型泛化能力的提升，如果不能，则不再继续生长子树。此时可能存在不同类别的样本同时存于结点中，按照多数投票的原则判断该结点所属类别。预剪枝对于何时停止决策树的生长有以下几种方法。

a. 当树到达一定深度的时候，停止树的生长。
b. 当到达当前结点的样本数量小于某个阈值的时候，停止树的生长。
c. 计算每次分类对测试集的准确度提升，当小于某个阈值的时候，不再继续扩展。

预剪枝具有思想直接、算法简单、效率高等特定，适合解决大规模问题。但如何准确地估计何时停止树的生长（即上述方法中的深度或阈值），针对不同问题会有很大差别，需要一定经验判断。且预剪枝存在一定局限性，有欠拟合的风险，虽然当前的划分会导致测试集准确率降低，但在之后的划分中，准确率可能会有显著上升。

后剪枝的核心思想是让算法生成一棵完全生长的决策树，然后从最底层向上计算是否剪枝。剪枝过程将子树删除，用一个叶子结点替代，该结点的类别同样按照多数投票的原则进行判断。同样地，后剪枝也可以通过在测试集上的准确率进行判断，如果剪枝过后准确率有所提升，则进行剪枝。相比于预剪枝，后剪枝方法通常可以得到泛化能力更强的决策树，但时间开销会更大。

常见的后剪枝方法包括错误率降低剪枝（Reduced Error Pruning, REP）、悲观剪枝（Pessimisic Error Pruning，PEP）、代价复杂度剪枝（Cost Complexiy Pruning，CCP）、最小误差剪枝（Minimum Error Pruning，MEP）、CVP（Critical Value Pruning）、OPP（Optimal Pruning）等方法，这些剪枝方法各有利弊，关注不同的优化角度。

参考文献

[1] Quinlan J R.Induction of decision trees. Machine Learning, 1986:1–356.
[2] Safavian S R , Landgrebe D . A survey of decision tree classifier methodology[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1991, 21(3):660–674.
[3] Nowozin S . Improved Information Gain Estimates for Decision Tree Induction[J]. Icml, 2012, 23(4):1293––1314.

[4] Chen X J , Zhang Z G , Tong Y . An Improved ID3 Decision Tree Algorithm[J]. Advanced Materials Research, 2014, 962-965:2842-2847.
[5] 张云涛. 数据挖掘原理与技术[M]. 电子工业出版社, 2004.