1 决策树是什么？

决策树(decision tree)是一种基本的分类与回归方法。我们可以这样理解，分类决策树模型是一种描述对实例进行分类的树形结构。决策树由结点(node)和有向边(directed edge)组成。结点有两种类型：内部结点(internal node)和叶结点(leaf node)。内部结点表示一个特征或属性，叶结点表示一个类。从结点引出的左右箭头就是有向边。而最上面的结点就是决策树的根结点(root node)。

2 我们可以把决策树看成一个 if-then 规则的集合：

由决策树的根结点(root node)到叶结点(leaf node)的每一条路径构建一条规则；路径上内部结点的特征对应着规则的条件，而叶结点的类对应着规则的结论。决策树的路径或其对应的 if-then 规则集合具有一个重要的性质：互斥并且完备。这就是说，每一个实例都被一条路径或一条规则所覆盖，而且只被一条路径或一条规则所覆盖。这里所覆盖是指实例的特征与路径上的特征一致或实例满足规则的条件。

3 使用决策树做预测需要以下过程：

收集数据：可以使用任何方法。比如想构建一个相亲系统，我们可以从媒婆那里，或者通过采访相亲对象获取数据。根据他们考虑的因素和最终的选择结果，就可以得到一些供我们利用的数据了。

准备数据：收集完的数据，我们要进行整理，将这些所有收集的信息按照一定规则整理出来，并排版，方便我们进行后续处理。

分析数据：可以使用任何方法，决策树构造完成之后，我们可以检查决策树图形是否符合预期。

训练算法：这个过程也就是构造决策树，同样也可以说是决策树学习，就是

构造一个决策树的数据结构。

测试算法：使用经验树计算错误率。当错误率达到了可接收范围，这个决策树就可以投放使用了。

使用算法：此步骤可以使用适用于任何监督学习算法，而使用决策树可以更好地理解数据的内在含义。

4 决策树分类标准

（1）究竟选择哪个特征更好些？这就要求确定选择特征的准则。直观上，如果一个特征具有更好的分类能力，（或者说，按照这一特征将训练数据集分割成子集，使得各个子集在当前条件下有最好的分类，）那么就更应该选择这个特征。信息增益就能够很好地表示这一直观的准则。

什么是信息增益呢？在划分数据集之后信息发生的变化称为信息增益，知道如何计算信息增益，我们就可以计算每个特征值划分数据集获得的信息增益，获得信息增益最高的特征就是最好的选择。信息增益是相对于特征而言的，信息增益越大，特征对最终的分类结果影响也就越大，我们就应该选择对最终分类结果影响最大的那个特征作为我们的分类特征。

在可以评测哪个数据划分方式是最好的数据划分之前，我们必须学习如何计算信息增益。集合信息的度量方式成为香农熵或者简称为熵(entropy)，这个名字来源于信息论之父克劳德·香农。在信息论与概率统计中，熵是表示随机变量不确定性的度量。熵越大，随机变量的不确定性就越大。如果待分类的事物可能划分在多个分类之中，则符号 xi 的信息定义为 ：

$$l(x_i) = -log_2 p(x_i)$$

其中 p(xi)是选择该分类的概率。

$$H = -\sum_{i=1}^{n} p(x_i)\, log_2 p(x_i)$$

其中 n 是分类的数目。熵越大，随机变量的不确定性就越大。

当熵中的概率由数据估计(特别是最大似然估计)得到时，所对应的熵称为经验熵(empirical entropy)。（什么叫由数据估计？比如有 10 个数据，一共有两个类别，A 类和 B 类。其中有 7 个数据属于 A 类，则该 A 类的概率即为十分之七。其中有 3 个数据属于 B 类，则该 B 类的概率即为十分之三。浅显的解释就是，这概率是我们根据数据数出来的。）我们定义贷款申请样本数据表中的数据为训练数据集 D，则训练数据集 D 的经验熵为 H(D)，|D|表示其样本容量，及样本个数。设有 K 个类 Ck, = 1,2,3,...,K,|Ck|为属于类 Ck 的样本个数，因此经验熵公式就可以写为 ：

$$H(D) = -\sum_{k=1}^{K} \frac{|C_k|}{|D|} \log_2 \frac{|C_k|}{|D|}$$

（2）选择特征，需要看信息增益。也就是说，信息增益是相对于特征而言的，信息增益越大，特征对最终的分类结果影响也就越大，我们就应该选择对最终分类结果影响最大的那个特征作为我们的分类特征。

在讲解信息增益定义之前，我们还需要明确一个概念，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，随机变量 X 给定的条件下随机变量 Y 的条件熵(conditional entropy)H(Y|X)，定义为 X 给定条件下 Y 的条件概率分布的熵对 X 的数学期望：

$$H(Y|X) = \sum_{i=1}^{n} p_i H(Y|X = x_i) \qquad p_i = P(X = x_i), i = 1,2,\cdots,n$$

同理，当条件熵中的概率由数据估计(特别是极大似然估计)得到时，所对应

的条件熵成为条件经验熵(empirical conditional entropy)。

明确了条件熵和经验条件熵的概念。接下来，让我们说说信息增益。前面也提到了，信息增益是相对于特征而言的。所以，特征 A 对训练数据集 D 的信息增益 g(D,A)，定义为集合 D 的经验熵 H(D)与特征 A 给定条件下 D 的经验条件熵 H(D|A)之差，即：

$$g(D,A) = H(D) - H(D|A)$$

设特征 A 有 n 个不同的取值{a1,a2, ··· ,an}，根据特征 A 的取值将 D 划分为 n 个子集{D1,D2, ··· ,Dn}，|Di|为 Di 的样本个数。记子集 Di 中属于 Ck 的样本的集合为 Dik，即 Dik = Di ∩ Ck，|Dik|为 Dik 的样本个数。于是经验条件熵的公式可以些为：

$$H(D|A) = \sum_{i=1}^{n} \frac{|D_i|}{|D|} H(D_i) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \sum_{k=1}^{K} \frac{|D_{ik}|}{|D_i|} \log_2 \frac{|D_{ik}|}{|D_i|}$$

5 决策树生成和修剪

我们已经学习了从数据集构造决策树算法所需要的子功能模块，包括经验熵的计算和最优特征的选择，其工作原理如下：得到原始数据集，然后基于最好的属性值划分数据集，由于特征值可能多于两个，因此可能存在大于两个分支的数据集划分。第一次划分之后，数据集被向下传递到树的分支的下一个结点。在这个结点上，我们可以再次划分数据。因此我们可以采用递归的原则处理数据集。

构建决策树的算法有很多，比如 C4.5、ID3 和 CART，这些算法在运行时并不总是在每次划分数据分组时都会消耗特征。由于特征数目并不是每次划分数据分组时都减少，因此这些算法在实际使用时可能引起一定的问题。目前我们

并不需要考虑这个问题，只需要在算法开始运行前计算列的数目，查看算法是否使用了所有属性即可。

决策树生成算法递归地产生决策树，直到不能继续下去未为止。这样产生的树往往对训练数据的分类很准确，但对未知的测试数据的分类却没有那么准确，即出现过拟合现象。过拟合的原因在于学习时过多地考虑如何提高对训练数据的正确分类，从而构建出过于复杂的决策树。解决这个问题的办法是考虑决策树的复杂度，对已生成的决策树进行简化。