研究生算法课课堂笔记

上课日期：2016 年 9 月 29 日　　第(1)节课

组长学号及姓名：马璁 1601214514

组员学号及姓名：秦嘉 1601214734

组员学号及姓名：宋勃宇 1601214519

一、 内容概要

本节课所学的内容包括以下几点：

1. 回顾上节课所学的知识：如何用树表示∧, ∨ ,¬ , XOR

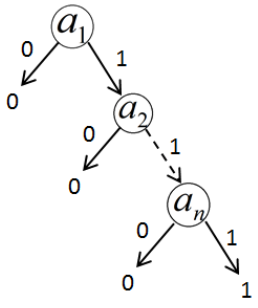
2. 建立决策树所需知识：熵(Entropy)和信息增益(Information Gain)

3. 建立决策树的方法与决策树算法的性质

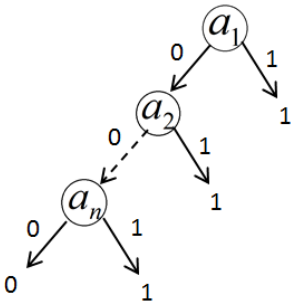
二、 详细内容

1. 回顾上节课所学的知识：如何用树表示∧, ∨, ¬ , XOR

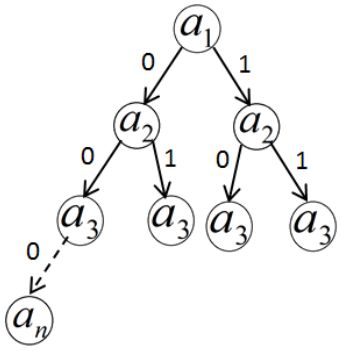
$a_1 \wedge a_2 \wedge \ldots \wedge a_n$



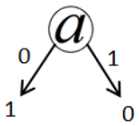
$a_1 \vee a_2 \vee \ldots \vee a_n$



$a_1 \oplus a_2 \oplus \ldots \oplus a_n$

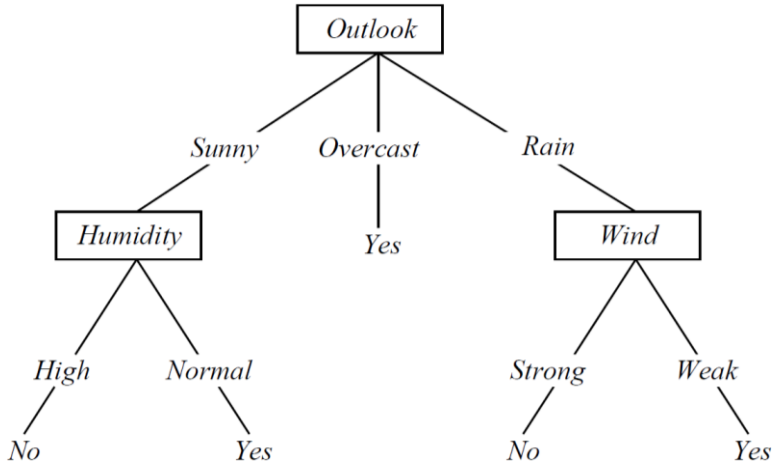


$\neg\, a$



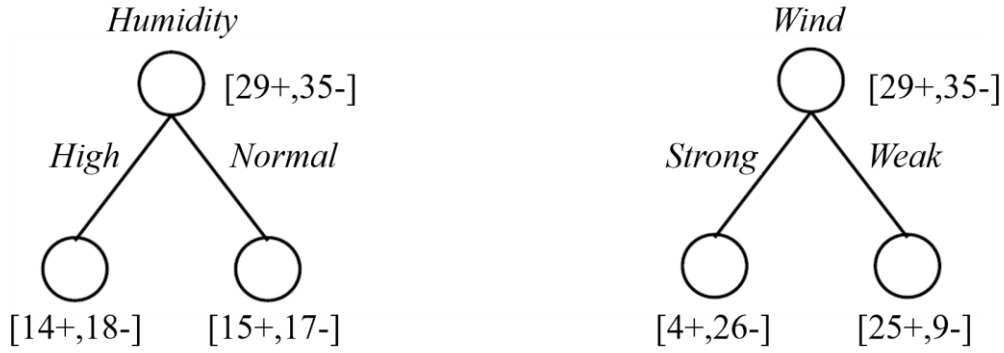
2. 决策树中所用到的信息论知识

这是一个典型的决策树 *PlayTennis*，根据外在条件判断是否去打网球。



老师之所以花 4-5 课时来讲信息论的知识，是为了通过熵、信息增益等信息来建立最好的决策树结构。对于决策树结构，如何判别决策树的好坏？

例如：训练样本集合 S 表示在不同天气条件下是否打网球，其中一共有 64 个样本包括 29 个正样本(去打网球)和 35 个负样本(不去打网球)，将 S 记为 *S=[29+,35-]*，有两个判别条件包括湿度(*Humidity*)和风力(*Wind*)，用这两个条件对样本 S 进行划分如下图所示。



由上图可知，对于湿度和风力两个判别条件，利用风力(Wind)对样本 S 进行分类比湿度(*Humidity*)所分类的效果明显，也就是说能将正负样本区分越明显的条件越能成为当前节点的判别条件。有两种极端情况，一种是该判别条件跟样本的结果毫无关联，这种情况下利用该条件判别出的正负样本的分布接近于原样本分布。与之相对应的另一种极端情况是该判别能将样本集的正负样本完全区分出来，即该判别条件的不同结果中的要么只有正样本，要么只有负样本。那么该怎

样选择最优的判别条件呢？下面需要回顾一下前几节课所讲的内容。

熵（*Entropy*）

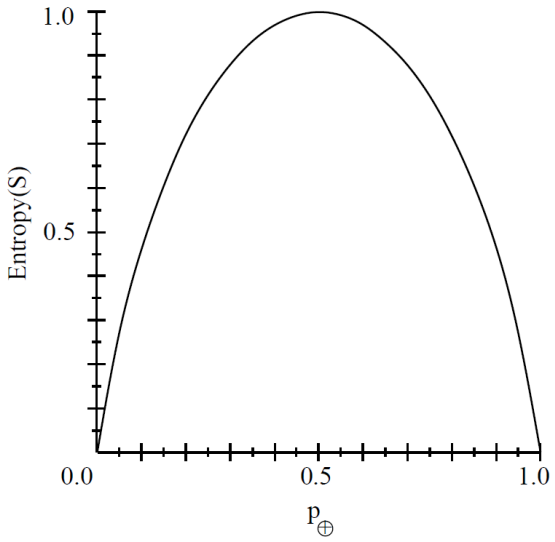
对于一个事件 S 的熵（*Entropy*）：

$$H(S)=\sum_{i=1}^{n}p(x_i)\log\frac{1}{p(x_i)}=-\sum_{i=1}^{n}p(x_i)\log p(x_i)$$

熵表现了信息的无序程度，当样本 *S* 所有成员属于同一类，其熵值为最小值即 *H(S)=0*，若 S 中的 *K* 种结果样本数量相等，其熵值为最大值即 *H(s)=logK*

对于布尔型分类（其样本 S 只分为正负两类）其熵的表示为：

$$H(S)=-p_+\log p_+-p_-\log p_-$$



上图为熵值与正样本所占比例的关系，*H(S)*取值范围*[0,1]*。

信息增益（*Information Gain*）

一个属性 *A* 相对于样本集合 *S* 的信息增益 *Gain(S,A)*的计算公式为：

$$Gain(S,A)=H(S)-\sum_{v\in Value(A)}\frac{|S_v|}{|S|}H(S_v)$$

其中 *Value(A)*是属性 *A* 的值域，*S* 是当前节点上的样本集合，S_v是 *S* 中属性 *A* 的取值为 *v* 的那些样本构成的子集合。

信息增益可以看做是由于一个判别条件所导致的原样本集合的期望熵降低的程度。上式中$H(S)$表示样本 *S* 的熵，对于一个确定的 *S* 其$H(S)$是定值。减号后面

是条件熵$H(S|A)$，表示在 A 条件下 S 的熵值。在上一章节讲到，选择最优的判别条件需要选择对样本区分程度明显的属性作为当前节点的划分属性，即其条件熵值越小说明该判决条件效益越大。

例如上一章节的打网球问题，S 包含 64 个样本*[29+,35-]*，在湿度条件下其中 14 个正样本和 18 个负样本属于 *Humidity=High*，15 个正样本和 17 个负样本属于 *Humidity=Normal*。按照 *Humidity* 属性分类 64 个样本得到的信息增益为：

$$Value(Humidity) = High, Normal$$

$$S = [29+, 35-]$$

$$S_{High} = [14+, 18-]$$

$$S_{Normal} = [15+, 17-]$$

信息增益：

$$Gain(S, Humidity) = H(S) - \sum_{V \in (High, Normal)} \frac{|S_v|}{|S|} H(S_v)$$

$$= H(S) - \frac{14+18}{29+35} H(S_{High}) - \frac{15+17}{29+35} H(S_{Normal})$$

$$= 0.9937 - \frac{32}{64} * 0.9887 - \frac{32}{64} * 0.9972 = 0.00075$$

其中：

$$H(S) = -\frac{29}{64} * \log\frac{29}{64} - \frac{35}{64} * \log\frac{35}{64} = 0.9937$$

$$H(S_{High}) = -\frac{14}{32} * \log\frac{14}{32} - \frac{18}{32} * \log\frac{18}{32} = 0.9887$$

$$H(S_{Normal}) = -\frac{15}{32} * \log\frac{15}{32} - \frac{17}{32} * \log\frac{17}{32} = 0.9972$$

相同方式得到根据 Wind 属性分类得到的信息增益：

$$S = [29+, 35-]$$

$$S_{High} = [4+, 26-]$$

$$S_{Normal} = [25+, 9-]$$

$$Gain(S, Wind) = H(S) - \sum_{v \in (Strong, Weak)} \frac{|S_v|}{|S|} H(S_v)$$

$$= 0.9937 - \frac{30}{64} * 0.5665 - \frac{34}{64} * 0.8338 = 0.2582$$

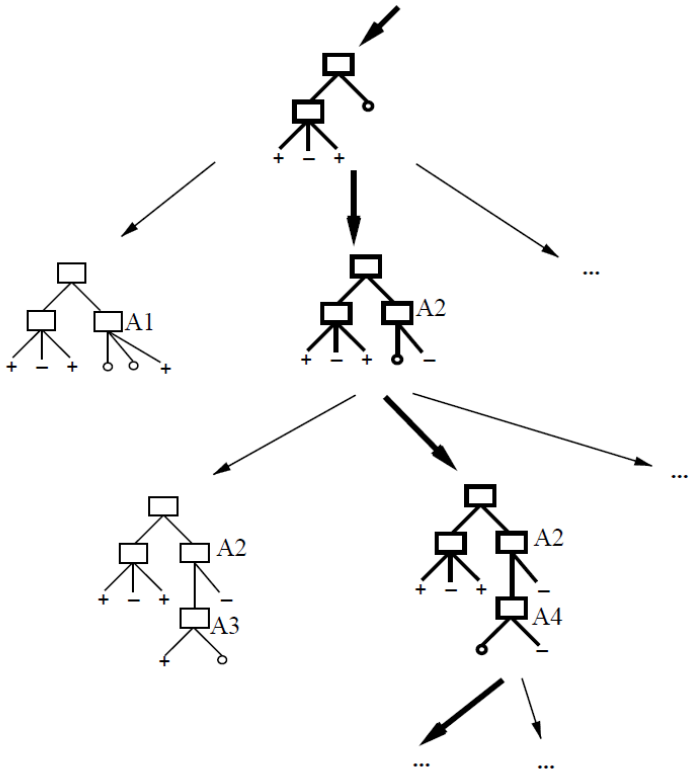
通过计算了两个不同属性：*Humidity* 和 *Wind* 的信息增益，最终 *Wind* 的信息增益大于 *Humidity* 的信息增益，所以采用 *Wind* 比采用 *Humidity* 作为分类属性更佳。

老师在课上提到过，在计算信息增益的过程中，被减数$H(S)$样本的熵值无论选择哪个属性来划分都是不变的，其减号后的分母$|S|$也是不变的，因此在选择最优属性时不用计算当前节点上样本的熵值，在计算划分后条件熵时也不用除以分母$|S|$。

3. 构建决策树

上一章节已经讲到了如何选择属性作为父节点，本章讨论如何搭建决策树搭建决策树所包含的思想有：

递归(*Recursion*)、贪心(*Greed*)和分治(*Divide and conquer*)，递归体现在建立决策树从父节点到子节点的过程，贪心体现在选择节点信息增益最大的属性，分治法体现在通过决策树的判别将样本分类进而减少样本数量。下图表示了决策树对节点的选择以及生长的示意图。

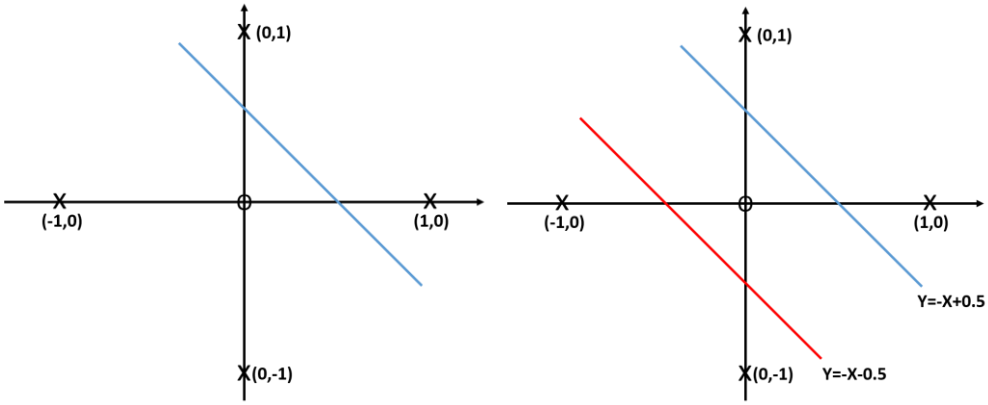


上图中分支下的+,-符号表示其分类已经完全分出了正负样本，o 中表示样本中仍然混杂着正负样本，需要进一步选择其他属性来对样本分类。所以在第一层右侧 o 中选择 A1 或 A2 属性进行分类，假设 A2 对样本的区分度更大，所以会选择 A2 作为其子节点进行延伸。后面对 A3 和 A4 的选择原理相同，以此类推完成决策树的搭建。但是这会出现一个问题，写递归算法的时候最怕不写清楚终止条件导致算法无限运行下去，决策树也是如此，当节点纯度达到 100%或样本个数小于某个阈值时不再继续划分。

信息增益 *Gain(S,A)*与互信息 *I(S；A)*的区别和联系？信息增益是衡量给定属性对样本划分能力的度量值，互信息是两个变量的关联程度，在决策树中这两者基本上是一个意思，因为属性与样本的关联性越强，该属性对样本的区分度也就越大。

4. 决策树的分类能力

决策树能否正确划分所有的训练样本？

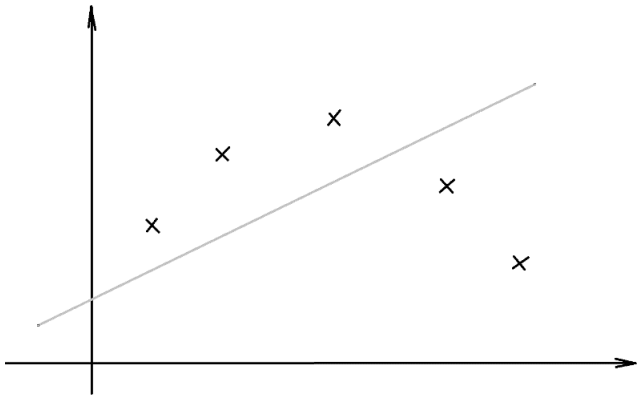


对于线性分类器不能保证对左图样本进行完全划分，如果像这样的两条线就可以对该数据集进行划分。

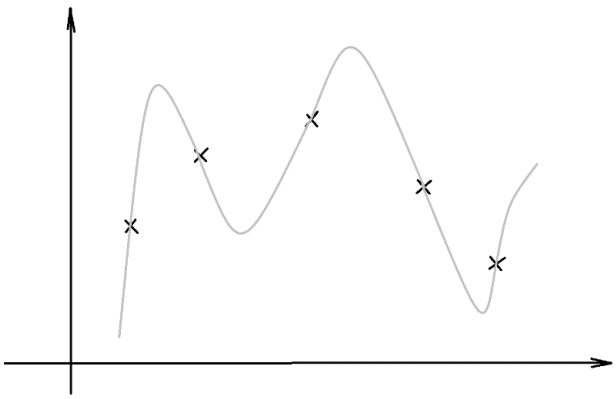
对于决策树来说，只要 X-Y 是个函数，则决策树必可分，无论是线性可分还是线性不可分。最笨的办法：考虑每个特征，构造一棵完全生长的二(K)叉树。但是对于 Stochastic function 来说，不能划分所有样本。所谓的 Stochastic function 是指函数的取值有一定的概率分布，属性全确定，但函数取值不确定。即在坐标系上的两个不同类别点完全重合，所以无法通过决策树划分。

5. 过拟合与欠拟合

High-bias：一个模型的表达能力有限，对数据分布的假设非常强，只能表达符合它假设的那些情况。比如线性回归，用直线拟合五个点，拟合结果不好：



High-variance:一个模型的表达能力非常强，对数据分布的假设很小。比如用 n-1 维多项式拟合 n 个点，一定能够完全拟合。但这样只能拟合训练集，不能用来做预测。对于多项式拟合来讲，多项式的次数越高，那么它的 *variance* 越大。



High-bias 和 *High-variance* 都不好。决策树属于 *High-variance*，表达能力很强，能表达非线性组合，只要是一个函数，则一定能表示出来。深度学习也属于 *High-variance*。

利用贪心算法构建决策树是否是最优的？

首先定义什么是最优：搜索深度低，用最少的层数将数据完全分开。答案是利用贪心算法构建决策树会陷入局部最优解，例如下节课笔记中老师举的异或的例子。在现实生活中，这种反例非常少，所以使用贪心算法构建决策树的效果一般是比较好的。

目前常用的决策树算法：

1）ID3 => C4.5 / C5.0

2）CART (Classification and Regression Tree)

ID3 算法特点：

1. Hypothesis Space 是完整的（Completed），模型能够刻画样本真实分布。
2. 没有 back-tracking。只能贪心地往下分，不能修改前面的。
3. Statistically-based search，划分属性是基于统计学的。即使样本中存在个别的噪音，只要样本数目足够大，是不会影响的。
4. 希望构造最短的树，在根节点信息增益最大的树。

   Occam’s Razor: 分的树太长，则模型的泛化能力很差，对新样本的预测能力差。