Machine Learning学习笔记（一）——决策树

决策树(Decision Trees)，一个非常经典的**监督式学习**案例。

**一、决策树基本介绍**

**1.决策树**

在建立决策树时候，计算机使用已分类的数据，建立一个具有预测功能的树形结构。

建立决策树有许多种算法,这里我只介绍**ID3算法**。

**2.核心算法**

**在建立决策树的时候，首先要选出哪一个属性作为最重要的属性，成为决策树的根节点**。这里将用到信息理论中的一个概念：熵（entropy）。具体计算使用下面的公式。

http://img.my.csdn.net/uploads/201301/28/1359379250_1364.png

对于这个公式我稍作解释。其中p(I)便是在集合S中属于类型I样本的比例。

如果S中所有的样本都属于I的话，P(I)便是1，所以http://img.my.csdn.net/uploads/201301/29/1359418925_5898.png，这也表示集合S是纯净的。

http://img.my.csdn.net/uploads/201301/29/1359419012_5083.png

公式2便是**信息增量，用于决定那个属性作为决策树的根节点。其中Gain越大的属性便是根节点**。这个公式可以从第二节的例子中得到解释。

## 二、决策树例子

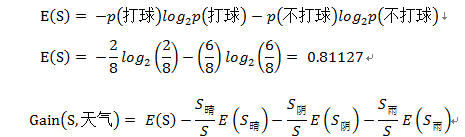


表2.1

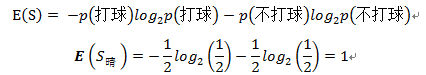
上表中包含八个样本，通过此表我们如何建立决策树？首先向计算出决策树的根节点，这样就要计算天气、身体状况和工作量的信息增益（Information Gain）。

取其中最大信息增益的条件。

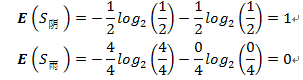
首先计算



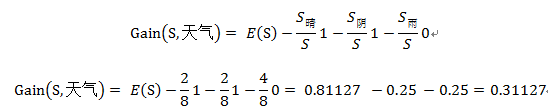
其中晴天的熵计算，其中天气为晴天的有两个样本，在这两个样本中，有两个行为结果。



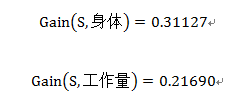
同样方式计算阴天和雨天的熵



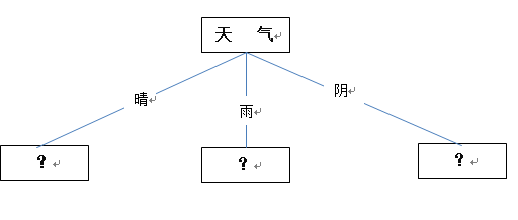
所以得到



同样的方法可以计算出另外两个(身体状况和工作量)的信息增益



三个条件计算完毕，得出天气和身体两个条件的信息增益是一样的，都比工作量大，我们这里就取天气作为根节点（当然也可以取身体）。我们已经找到了决策树的根节点，但是任务还没有完成。我们要开始建立三棵子树。



晴天条件下

|  |  |  |
| --- | --- | --- |
| **身体状况（Health）** | **工作量（Wordload）** | **行为（Behavior）** |
| 良好 | 轻松 | 打球 |
| 糟糕 | 繁重 | 不打球 |

表2.2

阴天条件

|  |  |  |
| --- | --- | --- |
| **身体状况（Health）** | **工作量（Wordload）** | **行为（Behavior）** |
| 良好 | 轻松 | 打球 |
| 糟糕 | 轻松 | 不打球 |

表2.3

雨天条件

|  |  |  |
| --- | --- | --- |
| **身体状况（Health）** | **工作量（Wordload）** | **行为（Behavior）** |
| 良好 | 轻松 | 不打球 |
| 良好 | 繁重 | 不打球 |
| 糟糕 | 轻松 | 不打球 |
| 糟糕 | 繁重 | 不打球 |

表2.4

 按照一开始介绍的方法，分别为上面三张表找决策树的根节点（信息增益最大的条件）。就这样不断递归建立。

Machine Learning学习笔记（二）——感知器学习算法

神经网络方面的知识（Neural Networks）。

**一、感知器学习算法基本介绍**

**1.神经网络**

就像进化计算，神经网络又是一个类似的概念。**神经网络由一个或者多个神经元组成。而一个神经元包括输入、输出和“内部处理器”。神经元从输入端接受信息，通过“内部处理器”将这些信息进行一定的处理，最后通过输出端输出。**

**2.感知器**

感知器（Perceptron），是神经网络中的一个概念，在1950s由Frank Rosenblatt第一次引入。

**3.单层感知器**

单层感知器（Single Layer Perceptron）是最简单的神经网络。它包含输入层和输出层，而输入层和输出层是直接相连的。

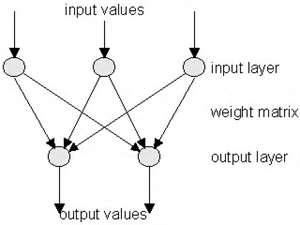


图1.1

图1.1便是一个单层感知器，很简单一个结构，输入层和输出层直接相连。

接下来介绍一下如何计算输出端。

http://img.my.csdn.net/uploads/201302/02/1359795126_4088.png

利用公式1计算输出层，这个公式也是很好理解。**首先计算输入层中，每一个输入端和其上的权值相乘，然后将这些乘机相加得到乘机和。对于这个乘机和做如下处理，如果乘机和大于临界值（一般是0），输入端就取1；如果小于临界值，就取-1。**

单层感知器其简单的特性，可以提供快速的计算。它能够实现逻辑计算中的**NOT、OR、AND等简单计算**。

但是对于稍微复杂的异或就无能无力。下面介绍的多层感知器，就能解决这个问题。

### 4.多层感知器

多层感知器（Multi-Layer Perceptrons），包含多层计算。

相对于单层感知器，输出端从一个变到了多个；输入端和输出端之间也不光只有一层，现在又两层:输出层和隐藏层。

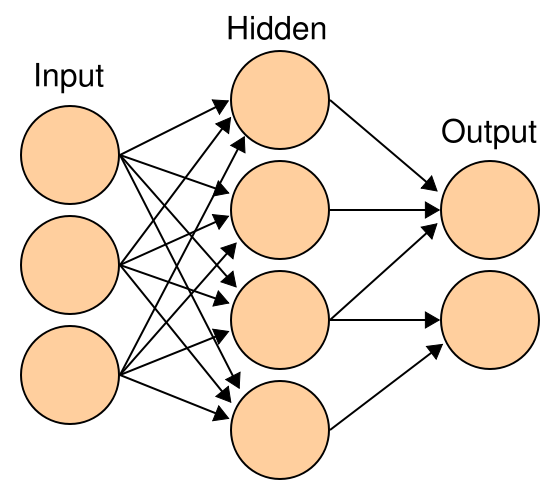


图2.2

图2.2就是一个多层感知器。

对于多层感知器的计算也是比较简单易懂的。首先利用公式1计算每一个。

**二、感知器学习算法**

**1.感知器学习**

其实感知器学习算法，就是利用第一节介绍的单层感知器。首先利用给的正确数据，计算得到输出值，将输出值和正确的值相比，由此来调整每一个输出端上的权值。

http://img.my.csdn.net/uploads/201302/02/1359795217_6393.png

公式2便是用来调整权值，首先 是一个“学习参数”，一般我将它设置成小于1的正数。T便是训练数据中的正确结果， http://img.my.csdn.net/uploads/201302/02/1359795230_8827.png便是第i个输入端的输入值，http://img.my.csdn.net/uploads/201302/02/1359795240_5666.png便是第i个输入端上面的权值。

2、总结：

感知器学习算法，是神经网络中的最简单的学习算法。

感知器学习算法，只要是利用了单层感知器。多层感知器主要是用于方向传播学习算法中。

附录：（待原理细节补充）

## 机器学习10大经典算法

1、**C4.5**

C4.5算法是机器学习算法中的一种分类决策树算法,其核心算法是ID3算法.  C4.5算法继承了ID3算法的优点，并在以下几方面对ID3算法进行了改进：

     1)用信息增益率来选择属性，克服了用信息增益选择属性时偏向选择取值多的属性的不足；

    2)在树构造过程中进行剪枝；

    3)能够完成对连续属性的离散化处理；

    4)能够对不完整数据进行处理。

C4.5算法有如下优点：产生的分类规则易于理解，准确率较高。其缺点是：在构造树的过程中，需要对数据集进行多次的顺序扫描和排序，因而导致算法的低效。

2、**The k-means algorithm即K-Means算法**

k-means algorithm算法是一个聚类算法，把n的对象根据他们的属性分为k个分割，k < n。它与处理混合正态分布的最大期望算法很相似，因为他们都试图找到数据中自然聚类的中心。它假设对象属性来自于空间向量，并且目标是使各个群组内部的均方误差总和最小。

3、**Support vector machines支持向量机**

支持向量机（Support Vector Machine），简称SV机（论文中一般简称SVM）。它是一种监督式学习的方法，它广泛的应用于统计分类以及回归分析中。支持向量机将向量映射到一个更高维的空间里，在这个空间里建立有一个最大间隔超平面。在分开数据的超平面的两边建有两个互相平行的超平面。分隔超平面使两个平行超平面的距离最大化。假定平行超平面间的距离或差距越大，分类器的总误差越小。一个极好的指南是C.J.C Burges的《模式识别支持向量机指南》。van der Walt和Barnard将支持向量机和其他分类器进行了比较。

4、**The Apriori algorithm**

Apriori算法是一种最有影响的挖掘布尔关联规则频繁项集的算法。其核心是基于两阶段频集思想的递推算法。该关联规则在分类上属于单维、单层、布尔关联规则。在这里，所有支持度大于最小支持度的项集称为频繁项集，简称频集。

5、**最大期望(EM)算法**

在统计计算中，最大期望（EM，Expectation–Maximization）算法是在概率（probabilistic）模型中寻找参数最大似然估计的算法，其中概率模型依赖于无法观测的隐藏变量（Latent Variabl）。最大期望经常用在机器学习和计算机视觉的数据集聚（Data Clustering）领域。

 参考：<http://blog.csdn.net/zouxy09/article/details/8537620>

6、**PageRank网页排名**

PageRank是Google算法的重要内容。2001年9月被授予美国专利，专利人是Google创始人之一拉里·佩奇（Larry Page）。因此，PageRank里的page不是指网页，而是指佩奇，即这个等级方法是以佩奇来命名的。

PageRank根据网站的外部链接和内部链接的数量和质量俩衡量网站的价值。PageRank背后的概念是，每个到页面的链接都是对该页面的一次投票，被链接的越多，就意味着被其他网站投票越多。这个就是所谓的“链接流行度”——衡量多少人愿意将他们的网站和你的网站挂钩。PageRank这个概念引自学术中一篇论文的被引述的频度——即被别人引述的次数越多，一般判断这篇论文的权威性就越高。

7、**AdaBoost**

Adaboost是一种迭代算法，其核心思想是针对同一个训练集训练不同的分类器(弱分类器)，然后把这些弱分类器集合起来，构成一个更强的最终分类器(强分类器)。其算法本身是通过改变数据分布来实现的，它根据每次训练集之中每个样本的分类是否正确，以及上次的总体分类的准确率，来确定每个样本的权值。将修改过权值的新数据集送给下层分类器进行训练，最后将每次训练得到的分类器最后融合起来，作为最后的决策分类器。

8、**kNN: k-nearest neighbor classification**

K最近邻(k-Nearest Neighbor，KNN)分类算法，是一个理论上比较成熟的方法，也是最简单的机器学习算法之一。该方法的思路是：如果一个样本在特征空间中的k个最相似(即特征空间中最邻近)的样本中的大多数属于某一个类别，则该样本也属于这个类别。

9、**Naive Bayes朴素贝叶斯**

在众多的分类模型中，应用最为广泛的两种分类模型是决策树模型(Decision Tree Model)和朴素贝叶斯模型（Naive Bayesian Model，NBC）。朴素贝叶斯模型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。但是实际上并非总是如此，这是因为NBC模型假设属性之间相互独立，这个假设在实际应用中往往是不成立的，这给NBC模型的正确分类带来了一定影响。在属性个数比较多或者属性之间相关性较大时，NBC模型的分类效率比不上决策树模型。而在属性相关性较小时，NBC模型的性能最为良好。

10、**CART:分类与回归树**

CART, Classification and Regression Trees。在分类树下面有两个关键的思想。第一个是关于递归地划分自变量空间的想法；第二个想法是用验证数据进行剪枝。