Wu You

2018 年 2 月 21 日

摘要

本次报告总结了 2017 年 12 月 26 日至 2018 年 1 月 20 日期间的学习研究进度。主要分为两部分内容，分别为机器学习理论算法以及机器学习用到的工具。理论方面主要学习了三种常用的学习算法，决策树学习、贝叶斯学习以及人工神经网络，学习了以决策树算法实现的工具，并尝试将其应用与数字图片识别，效果较好；针对文档分类问题，实现了简单的朴素贝叶斯算法，并实验用于文本分类；人工神经网络方面主要学习了前馈网络部分的内容。工具使用的是 python 的 scikit-learn 机器学习类库。

关键词：机器学习；决策树；贝叶斯；神经网络；scikit-learn

目录

1. 机器学习基础概念

机器学习基础概念较多,

1.

2. 决策树算法

决策树算法是使用较为广泛的一种分类算法,因为实现过程较为简单。它主要用于逼近离散值函数,且对噪声有良好的健壮性,能够学习到含有析取表达式的假设。

2.1. 决策树的构建过程

决策树算法通过结合训练样例的目标值,对训练样例的各个属性依次分类,每个属性为一个节点,若节点中含有目标属性值不同的样例,则进一步进行划分,最终形成一个树形结构,该树可为二叉树或多叉树。构建树的首个问题是选择最佳的初始分类属性,也就是根节点最佳属性的选择。

2.1.1. 信息增益

对于概念学习,信息熵的定义如下:

$$Entropy(S) = -p_{\oplus} \cdot log_2 p_{\oplus} - p_{\ominus} \cdot log_2 p_{\ominus}$$

其中 $p_{\oplus}$ 和 $p_{\ominus}$ 分别为被该属性分为正例和反例的样本所占比例。进一步地,对于目标属性有 c 个取值的概念,信息熵为:

$$Entropy(S) = \sum_{i=1}^{c} -p_i \cdot log_2 p_i$$

对任意属性,其信息熵最大值为 $log_2 c$ 最小值为 0。信息熵描述了样本的均一性。有了信息熵,之后可以对信息增益进行定义,对样本集合 S,某一属性 A 的信息增益定义如下:

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

该属性刻画了属性 A 对样本 S 的分类效果,信息增益越大,表明样例信息熵降低越多,也就意味着分类后的样本更加一致,在进行最佳分类属性的选择时,以信息增益取最大值为标准。

2.1.2. 决策树修剪

在对样本的学习过程中，由于噪声或特殊样例的存在，通常会造成过拟合的情况发生，即该树在训练样本集合上表现良好，但在对新的实例进行分类时准确率欠佳（泛化精度低），为提高泛化精度，需要额外的处理，主要有两种方式提高泛化精度：

- 提前停止树的增长
- 对树进行后修剪

对于以上两种方法，有一个共同的问题：确定其遵循的准则，也就是何时停止或对树修剪到何种程度，对此问题有以下几种解决方案：

- 用新的样例来评估效果
- 对增长或修剪操作的效果进行估计
- 用编码来衡量

其中, 第一种方法最为常用, 我们可以将训练样例分为训练集和验证集, 比例为 1:1 或 2:1, 但验证集要足够大。（k-折交叉验证与此类似，此处应为 2-折交叉验证）下面是两种树的修剪方法：

1. 错误率降低修剪
2. 规则后修剪

错误率降低修剪是将以某一节点为根的树全部移除，并将属于该子树的最常见目标属性值赋予该节点，但仅在可以降低错误率时进行此操作。规则后修剪方法首先令树生长到尽可能拟合训练数据，之后将树转化为等价的规则集合，为根节点到每个叶节点的路径创建规则（$\wedge$）, 之后对这些规则进行泛化以提高其精度，这一方法使得修剪更为灵活，并且对人来说，规则更加容易理解。

2.1.3. 一些特殊属性的处理

连续值属性

决策树在处理离散值时，可以将其取值范围划分为多个区间，从而像对待离散值属性一样处理该属性。

有大量取值的属性

例如日期这类属性，拥有大量的不同属性值，假设在某一节点以日期为 1 号对样例进行划分，会使得样例被划分为一大一小两部分，此时该属性的信息增益将大于其他属性，但实际上该属性

分类效果很差，此时需要用其他标准来代替信息增益，信息增益比率是一个可用的标准。首先定义分裂信息 (Splitinformation)：

$$splitinformation(S,A) = \sum_{i=1}^{c} \frac{|S_i|}{S} \cdot log\frac{|S_i|}{S}$$

信息增益比率的定义如下:

$$Gainratio(S,A) = \frac{GainS,A}{splitinformation(S,A)}$$

实际上，分裂信息就是 S 关于 A 的熵，当 A 的取值数量较多时，其分裂信息也较大，信息增益比率的值会相应减小，这相当于对取值较多的属性进行了惩罚。

处理缺少的属性

对于缺少某属性的样例，可赋予其最常见值，或按照已观察到的该属性取值比例对这些样例进行赋值。

处理不同代价的属性

3. 贝叶斯方法

贝叶斯方法利用先验知识和观察数据一同决定假设的最终概率，可以对假设做不确定性的预测，该方法也可用于衡量其他算法的最优决策。

3.1. 贝叶斯法则

对某一假设 h 和训练样例 D，有如下的贝叶斯公式：

$$P(h|D) = \frac{P(D|h) \cdot P(h)}{P(D)}$$

其中，$P(h)$ 是 h 的先验概率，基于现有知识得到，$P(D|h)$ 是在我们认为 h 成立的情况下，观察到 D 的概率，而 $P(D)$ 是 D 的先验概率，等式左边的 $P(h|D)$ 为假设 h 的后验概率，即观察到

D 时，h 成立的概率。极大后验概率假设（MAP）定义为使后验概率最大的假设：

$$\begin{aligned} h_{MAP} &= \underset{h \in H}{argmax}\, P(h|D) \\ &= \underset{h \in H}{argmax}\, P(D|h)P(h) \end{aligned}$$

有时候我们会给每个假设赋予相同的先验概率，这时，我们只需要根据 $P(D|h)$ 来决定最大可能的假设，$P(D|h)$ 常称为 h 成立时样本 D 的似然度，使该概率最大的假设称为极大似然假设（ML）：

$$h_{ML} = \underset{h \in H}{argmax}\, P(D|h)$$

3.2. 贝叶斯理论的应用

对于具有一组属性 $a_1, a_2, ..., a_n$ 的实例来说，其极大后验假设为：

$$h_{MAP} = \underset{(}{argmax}\, v_j \in V)P(v_j|a_1, a_2, a_3, ..., a_n)P(v_j)$$

朴素贝叶斯分类器基于以下假设：各个属性值的取值是互相独立的。因此，其极大后验假设可以进一步表示为：

$$\begin{aligned} h_{MAP} &= v_{NB} \\ &= \underset{v_j \in V}{argmax}\, P(v_j|a_1, a_2, a_3, ..., a_n)P(v_j) \\ &= \underset{v_j \in V}{argmax} \prod_{i=1}^{n} P(a_i|v_j)P(v_j) \end{aligned}$$

$$P(a_i|v_j) = \frac{n_c + mp}{n + m}$$