习题三

141210016, 刘冰楠, bingnliu@outlook.com

2017 年 7 月 26 日

1 [30pts] Decision Tree Analysis

决策树是一类常见的机器学习方法，但是在训练过程中会遇到一些问题。

(1) **[15pts]** 试证明对于不含冲突数据 (即特征向量完全相同但标记不同) 的训练集，必存在与训练集一致 (即训练误差为 0) 的决策树;

(2) **[15pts]** 试分析使用 “最小训练误差” 作为决策树划分选择的缺陷。

Solution.

1.1 Problem (1)

记号说明：设共有 k 个特征属性，对每个示例 $\mathbf{x}^{(j)}$ $(1 \le j \le n)$，用 $x_1^{(j)}, x_2^{(j)}, x_3^{(j)}, \ldots, x_k^{(j)}$ 来表示其各个特征属性的取值，其中 $x_1^{(j)}, x_2^{(j)}, x_3^{(j)}, \ldots, x_\ell$ 为离散属性，$x_{\ell+1}^{(j)}, x_{\ell+2}^{(j)}, x_{\ell+3}^{(j)}, \ldots, x_k^{(j)}$ 为连续属性（可能只存在离散或连续属性）。

1.1.1 数学归纳法证明

如果只有一个示例，按其标记划分即可得到训练误差为 0 的决策树；假设目前有 n 个示例 $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \mathbf{x}^{(3)}, \ldots, \mathbf{x}^{(n)}$，和在这 n 个示例上训练误差为 0 的决策树 T_n，添加一个新的样例 $\mathbf{x}^{(n+1)}$，满足 $\mathbf{x}^{(n+1)}$ 与已有的 n 个示例不冲突，那么只需按如下步骤修正已有的决策树，即可得到在这 $n+1$ 个示例上训练误差为 0 的决策树 T_{n+1}：

1. 用决策树 T_n 对 $\mathbf{x}^{(n+1)}$ 进行分类，根据分类结果的正确性，分两种情况：
   (a) 若 $\mathbf{x}^{(n+1)}$ 分类正确，则无需对 T_n 进行修改，已经达成目标；
   (b) 若 $\mathbf{x}^{(n+1)}$ 分类错误，根据是否存在连续属性分两种情况：
      i. 若存在连续属性，则任选一连续属性 $x_p^{(n+1)}$。若 $x_p^{(n+1)}$ 是最大值，且第二大的值是 x_p^i 则在当前叶节点增加判断 $x_p \le x_p^{(i)}$?，即可将 $\mathbf{x}^{(n+1)}$ 单独划出来；若 $x_p^{(n+1)}$ 是最小值，且第二小的值是 x_p^i 则在当前叶节点增加判断 $x_p \le x_p^{(n+1)}$?，即可将 $\mathbf{x}^{(n+1)}$ 单独划出来；若 $x_p^{(n+1)}$ 既不是最大值也不是最小值，且从小到大排列，相邻的两个属性是 $x_p^{(i)} < x_p^{(n+1)} < x_p^{(j)}$，则增加 $x_p \le x_p^{(n+1)}$?, $x_p \le x_p^{(i)}$? 即可将 $\mathbf{x}^{(n+1)}$ 单独划分入一个叶节点。得到新的决策树 T_{n+1}.

ii. 若不存在连续属性，则逐一使用每一个尚未被使用（必然存在，否则有 $\mathbf{x}^{(n+1)}$ 和与其同在一个叶节点的示例特征属性完全相同，但标记不同，与题中假设矛盾）的离散属性 x_q 进一步分类，必然可以将 $\mathbf{x}^{(n+1)}$ 与该叶节点的其他示例分开（仍然是因为不存在冲突数据）。分开后，只需对新产生的叶节点按照其中示例的（共同的）标记来分类，得到新的决策树 T_{n+1}.

由数学归纳法知，对任何大小的不含冲突数据的训练集，必存在与训练集一致的决策树。

□

1.1.2 直接构造法证明

设对于当前训练集，特征属性 x_i 有 n_i 个取值（对于离散属性即为不同类别的个数，对于连续属性则为不同实值的个数）。

构造一颗共有 k 层的决策树，对于第 i 层的**每个**节点，按特征属性 x_i 分为 n_i 支，最后我们得到了一个共有 $\prod_{i=1}^{k} n_i$ 个叶节点的（超大）决策树。由于不同的特征向量最多有 $\prod_{i=1}^{k} n_i$ 种，因此这棵树可以将具有不同特征向量的示例划分到不同叶节点中。又由题知特征向量相同的节点标记也相同，所以只需对所有非空叶节点按照其中示例的（共同的）标记来分类，即可得到在当前训练集上误差为 0 的决策树。

□

1.2 Problem (2)

主要缺陷在于**过拟合**风险太大。下面详述。

1.3 使用课本中的递归算法，无法选择划分属性

如果题目中 “划分选择” 是指决策树算法中如何定义 “最优” 划分属性，那么训练未完成的时候，还未给结点进行标记，因而得不到每个样本的预测类别，也就**没法**计算训练样本误差。也就是说，课本图 4.2 伪代码所描述的递归训练方法，与 “最小训练误差” 这个划分选择标准是不兼容的。

1.4 使用类似预剪枝中的方法，则可以划分并分析

预剪枝中在判断是否剪枝时，比较了剪枝前后的验证集误差。其中计算验证集误差时，把结点所有样本标记为占比最大的那一类。如果用这样的方法，就可以以 “最小训练集误差” 作为划分选择：

选择最优划分属性时，对属性集 A 中的每个属性，分别作为划分属性，**划分后的每个结点直接成为叶节点**，并标记为结点样本占比最大的那一类，然后计算训练集误差，并选择误差最小的那个属性作为划分属性。

对某结点根据某属性进行划分后，训练集误差**只可能减小或不变**（划分后一个结点会变多个，如果这多个结点仍标记为之前的类别，则误差不变，反之则可能减小），在 “最小化训练样本误差” 的驱动下，结点划分过程将不断重复，使得生成的决策树学到了不该学的特征—训练样本特有的特征，从而导致**过拟合**。

例如第一问所答，只要不存在数据冲突，必然能得到训练集误差为 0 的决策树，但它的泛化性能不太可能好，所以没有意义。而在多变量决策树，由于非平行边界甚至非线性分类器的引入，只最小化训练集误差更是会促使决策树把训练集特有的特征学得**淋漓尽致**，从而泛化性能更差。

这也是剪枝判断时，使用**验证集**误差而不是训练集的原因。

2 [30pts] Training a Decision Tree

考虑下面的训练集：共计 6 个训练样本，每个训练样本有三个维度的特征属性和标记信息。详细信息如表1所示。

请通过训练集中的数据训练一棵决策树，要求通过 “信息增益”(information gain) 为准则来选择划分属性。请参考书中图 4.4，给出详细的计算过程并画出最终的决策树。

表 1: 训练集信息

序号	特征 **A**	特征 **B**	特征 **C**	标记
1	0	1	1	0
2	1	1	1	0
3	0	0	0	0
4	1	1	0	1
5	0	1	0	1
6	1	0	1	1

Solution.

以信息增益为划分选择标准，等价于选择能使划分后加权信息熵最小的属性。加权信息熵即为：

$$\sum_{v=1}^{V} \frac{|D^v|}{|D|} \mathrm{Ent}(D^v) \tag{1}$$

2.1 训练过程

递归深度 =0: 根节点包含所有样本，不满足停止条件，则按信息增益法则寻找最优划分属性：

表 2: 选择根结点的划分属性

属性	0 分支两类样本占比	0 分支权重	1 分支两类样本占比	1 分支权重	加权信息熵
A	2/3, 1/3	3/6	1/3, 2/3	3/6	$\log_2 3 - 2/3$
B	1/2, 1/2	2/6	1/3, 2/4	4/6	1
C	1/3, 2/3	3/6	2/3, 1/3	3/6	$\log_2 3 - 2/3$

由表2选择 C 作为根节点划分属性 (此处 C 和 A 信息增益相同，需引入选择偏好，见下方)，得到两个分支 C=0 和 C=1，均不为空，则递归继续划分。

递归深度 =1: 对于根节点的 C=0 分支，不满足返回条件，则按信息增益法则寻找最优划分属性：

表 3: 选择 C=0 分支结点的划分属性

	0 分支两类样本占比	0 分支权重	1 分支两类样本占比	1 分支权重	加权信息熵
A	1/2, 1/2	2/3	0, 1	1/3	2/3
B	1, 0	1/3	0, 1	2/3	0

由表3选择 B 作为根节点划分属性，得到两个分支 B=0 和 B=1，均不为空，则递归继续划分。

递归深度 =2: 对于分支 B=0，满足递归返回条件中的情形 (1): 结点所有样本全属于同一类别，返回。

递归深度 =2: 对于分支 B=1，满足递归返回条件中的情形 (1): 结点所有样本全属于同一类别，返回。

递归深度 =1: 对于根节点的 C=1 分支，不满足返回条件，则按信息增益法则寻找最优划分属性：

表 4: 选择 C=1 分支结点的划分属性

	0 分支两类样本占比	0 分支权重	1 分支两类样本占比	1 分支权重	加权信息熵
A	1, 0	1/3	1/2, 1/2	2/3	2/3
B	0, 1	1/3	1, 0	2/3	0

由表4选择 B 作为根节点划分属性，得到两个分支 B=0 和 B=1，均不为空，则递归继续划分。

递归深度 =2: 对于分支 B=0，满足递归返回条件中的情形 (1): 结点所有样本全属于同一类别，返回。

递归深度 =2: 对于分支 B=1，满足递归返回条件中的情形 (1): 结点所有样本全属于同一类别，返回。

至此所有递归调用都已返回，决策树训练完成。

2.2 最终的决策树

2.3 多种划分属性信息增益相同时的讨论

对于较大的训练样本集，不同划分属性的信息增益相同的可能性很低。但还是有可能，见表2，本题在选择根节点的划分属性时，A 和 C 的信息增益相同。

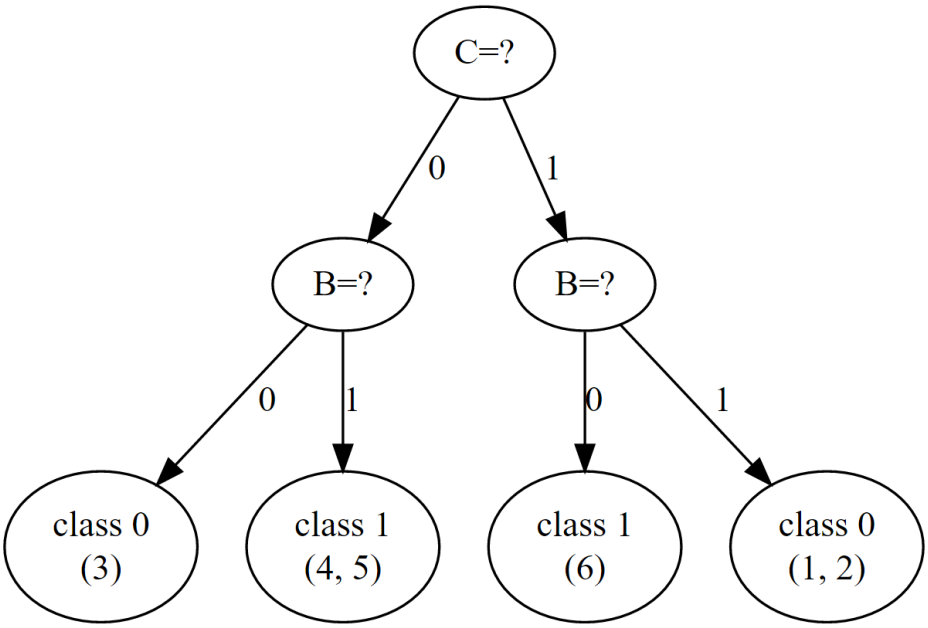

图 1: 根据训练样本训练生成的决策树.

如果信息增益相同，就会出现课本第一章所讲的**版本空间**问题，即多个不同的决策树满足算法要求。因而此时需引进**选择偏好**，如在上文选择根节点划分属性时，选择了最后一个特征属性。

其他可能的选择偏好如：在信息增益相同的划分属性中，选择增益率最大的，如果增益率相同，则按照属性在内存中存储的下标，选下标最小的。

3 [40pts] Back Propagation

单隐层前馈神经网络的误差逆传播 (error BackPropagation，简称 BP) 算法是实际工程实践中非常重要的基础，也是理解神经网络的关键。

请编程实现 BP 算法，算法流程如课本图 5.8 所示。详细编程题指南请参见链接：`http://lamda.nju.edu.cn/ml2017/PS3/ML3_programming.html`

在实现之后，你对 BP 算法有什么新的认识吗？请简要谈谈。

Solution.

3.1 编程作业

见 main.py

3.2 对 BP 算法的新的认识

3.2.1 编程实现有助于数学理解

课本中的梯度更新部分，公式下标很多，看起来很容易晕。而在实现的过程中，需要仔细地考虑每一个变量代表的是标量还是矢量，是几维，需不需要转置等，写下来后对公式理解清晰了很多。

3.2.2 人工智能就是工具

周老师上课讲过，我们受神经元启发，做神经网络、人工智能，但最终根基都是数学证明，我们也不去担心它会不会有了思想、出什么乱子；就像仿照飞鸟做飞机的人，最终根基是空气动力学，不会去担心飞机上了天会不会和老鹰谈恋爱。

我从头到位把 BP 算法一气呵成写完，第一次运行的时候，虽然很慢，但无论 syntax 还是逻辑都没有 bug，看到每一轮的训练误差真的在降低，验证集精度也有 90%+，就突然理解了周老师上课的话。BP 神经网络里面没有神奇的神经元，什么激活什么的，只是一个个向量加减乘除，判断循环。

附加题 [30pts] Neural Network in Practice

在实际工程实现中，通常我们会使用已有的开源库，这样会减少搭建原有模块的时间。因此，请使用现有神经网络库，编程实现更复杂的神经网络。详细编程题指南请参见链接：`http://lamda.nju.edu.cn/ml2017/PS3/ML3_programming.html`

和上一题相比，模型性能有变化吗？如果有，你认为可能是什么原因。同时，在实践过程中你遇到了什么问题，是如何解决的？

Solution.

3.3 编程作业

见 main_library.py

3.4 模型性能变化

对应于第三题实现的标准 bp，相当于设置 batch_size=1，使用该设置在 fit 时非常缓慢，因为 keras 默认每完成一个 batch 都要计算 accuracy，于是调整 batch_size=10，经过同样的轮数（24 轮）得到结果为：training data acc: 98.87%, test data acc: 93.50%.

而我自己实现的标准 BP 精度为 train data accu:0.962, test data accu:0.919

相比之下调包结果很好，一方面可能是 batch_size 不同造成的，另一方面可能是包中在梯度下降步长选择等小地方有更好的 trick 导致。

3.5 实践中遇到的问题

- y 的 data 要进行 one hot encoding（从 int 变为 01 编码）
- 只根据编程指南给出的参数调用函数是不够的其他必须参数如 input_dim 在 keras 的 tutorial 里有介绍
- 针对目前的作业，不需要学习 tensorflow，直接用 keras 写就可以，可以从很高的抽象层次写，keras 会自动调用 tensorflow 作为 backend 计算