机器学习导论
习题三

141242006, 袁帅, 141242006@smail.nju.edu.cn

2017 年 4 月 24 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) For all training sets(without contradictory samples) with n samples, we can prove that there must exist one decision tree corresponding to this dataset with 0 training errors, by mathematical induction as follow:

Proof. Basis: When $n = 1$, such decision tree could obviously be built by simply adding one node.

Induction Hypothesis: When $n \geq 2$, suppose the statement holds for all $1 \leq k < n$.

Induction Steps: Consider training sets $D = (\boldsymbol{X}, \boldsymbol{y})$ with n samples($n \geq 2$). We can choose a feature for split by finding such feature a on which the samples take more than one value (i.e., $\exists \boldsymbol{x}_1, \boldsymbol{x}_2 \in \boldsymbol{X}$, s.t. $\boldsymbol{x}_1^{(a)} \neq \boldsymbol{x}_2^{(a)}$, where $\boldsymbol{x}^{(a)}$ denotes the value of $\boldsymbol{x}$ on feature a). Note that if such a doesn't exist, all samples in the training sets are identical, yielding a trivial case.

Now we split the dataset D by feature a into p sub-datasets $D_1, D_2, ..., D_p$, the number of samples of which are $n_1, n_2, ..., n_p$, respectively. Because $1 \leq n_i < n$ for all $1 \leq i \leq p$, according to the induction hypothesis, all these sub-datasets must have their own corresponding decision trees, and therefore by linking those trees under one single node, we generate a decision tree, which takes a as the first split, for dataset D. $\square$

(2) If we split the tree by minimizing training error, the model would suffer from overfitting. If all splits are based on training error, it does not generalize well with other data.

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. We train the decision tree, split by information gain $\mathrm{Gain}(D, a)$.

(1) For training set D and feature set $F = \{A, B, C\}$, $|\mathcal{Y}| = 2$, $p_0 = \frac{1}{2}$, $p_1 = \frac{1}{2}$, and therefore information entropy $\mathrm{Ent}(D) = -\sum_{k=0}^{1} p_k \log_2 p_k = 1$.

If D is split by feature A, the resulting subsets $D^0 = \{1, 3, 5\}$, $D^1 = \{2, 4, 6\}$, $\mathrm{Ent}(D^0) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.9183$, $\mathrm{Ent}(D^1) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.9183$, so the information gain $\mathrm{Gain}(D, A) = \mathrm{Ent}(D) - \sum_{k=0}^{1}\frac{|D^k|}{|D|}\mathrm{Ent}(D^k) = 1 - 0.9183 = 0.0817$.

If D is split by feature B, the resulting subsets $D^0 = \{3, 6\}$, $D^1 = \{1, 2, 4, 5\}$, $\mathrm{Ent}(D^0) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$, $\mathrm{Ent}(D^1) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$, so the information gain $\mathrm{Gain}(D, A) = \mathrm{Ent}(D) - \sum_{k=0}^{1}\frac{|D^k|}{|D|}\mathrm{Ent}(D^k) = 1 - 1 = 0$.

If D is split by feature C, the resulting subsets $D^0 = \{3, 4, 5\}$, $D^1 = \{1, 2, 6\}$, $\mathrm{Ent}(D^0) = -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = 0.9183$, $\mathrm{Ent}(D^1) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.9183$, so the information gain $\mathrm{Gain}(D, A) = \mathrm{Ent}(D) - \sum_{k=0}^{1}\frac{|D^k|}{|D|}\mathrm{Ent}(D^k) = 1 - 0.9183 = 0.0817$.

So we will split D by feature A, yielding $D_1 = \{1, 3, 5\}$ and $D_2 = \{2, 4, 6\}$.

(2) For $D_1 = \{1, 3, 5\}$ and $F_1 = \{B, C\}$, $|\mathcal{Y}| = 2$, $p_0 = \frac{2}{3}$, $p_1 = \frac{1}{3}$, and therefore information entropy $\mathrm{Ent}(D_1) = -\sum_{k=0}^{1} p_k \log_2 p_k = 0.9183$.

If D_1 is split by feature B, the resulting subsets $D_1^0 = \{3\}$, $D_1^1 = \{1, 5\}$, $\mathrm{Ent}(D_1^0) = 0$, $\mathrm{Ent}(D_1^1) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$, so the information gain $\mathrm{Gain}(D_1, B) = \mathrm{Ent}(D_1) - \sum_{k=0}^{1}\frac{|D_1^k|}{|D_1|}\mathrm{Ent}(D_1^k) = 0.9183 - 0.6667 = 0.2516$.

If D_1 is split by feature C, the resulting subsets $D_1^0 = \{3, 5\}$, $D_1^1 = \{1\}$, $\mathrm{Ent}(D_1^0) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$, $\mathrm{Ent}(D_1^1) = 0$, so the information gain $\mathrm{Gain}(D_1, C) = \mathrm{Ent}(D_1) - \sum_{k=0}^{1}\frac{|D_1^k|}{|D_1|}\mathrm{Ent}(D_1^k) = 0.9183 - 0.6667 = 0.2516$.

So we will split D_1 by feature B, yielding $D_3 = \{3\}$ and $D_4 = \{1, 5\}$.

(3) For $D_2 = \{2, 4, 6\}$ and $F_2 = \{B, C\}$, $|\mathcal{Y}| = 2$, $p_0 = \frac{1}{3}$, $p_1 = \frac{2}{3}$, and therefore information entropy $\mathrm{Ent}(D_1) = -\sum_{k=0}^{1} p_k \log_2 p_k = 0.9183$.

If D_2 is split by feature B, the resulting subsets $D_2^0 = \{6\}$, $D_2^1 = \{2, 4\}$, $\mathrm{Ent}(D_2^0) = 0$, $\mathrm{Ent}(D_2^1) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$, so the information gain $\mathrm{Gain}(D_2, B) = \mathrm{Ent}(D_2) - \sum_{k=0}^{1}\frac{|D_2^k|}{|D_2|}\mathrm{Ent}(D_2^k) = 0.9183 - 0.6667 = 0.2516$.

If D_2 is split by feature C, the resulting subsets $D_2^0 = \{4\}$, $D_2^1 = \{2, 6\}$, $\mathrm{Ent}(D_2^0) = 0$, $\mathrm{Ent}(D_2^1) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$, so the information gain $\mathrm{Gain}(D_2, C) = \mathrm{Ent}(D_2) - \sum_{k=0}^{1}\frac{|D_2^k|}{|D_2|}\mathrm{Ent}(D_2^k) = 0.9183 - 0.6667 = 0.2516$.

So we will split D_2 by feature B, yielding $D_5 = \{6\}$ and $D_6 = \{2, 4\}$.

(4) For $D_4 = \{1, 5\}$ and $F_4 = \{C\}$, the last split would by feature C.

(5) For $D_6 = \{2, 4\}$ and $F_6 = \{C\}$, the last split would by feature C.

Therefore, the consequent decision tree could be shown as below:

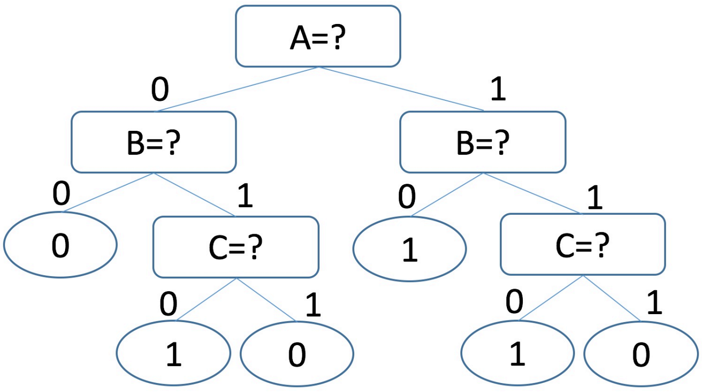

Figure 1: Decision tree for training set D

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. I implemented the code in MATLAB, yielding an accuracy around 94%. The core thing of BP neural networks is nothing but a gradient descent approach, in which the gradient could be vividly calculated and presented.

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. With the help of Keras Documentation[1], I finished the code in Python. The model training process is much faster than the previous model, while no significant progress is found in accuracy. The training speed increases a lot because of the proper implementation and compiling optimization of library functions; the accuracy is not improved because our previous code already works well enough: 94% accuracy is arguably high for a MNIST dataset with only 3000 training samples.

Reference

[1] *Keras Documentation*. `https://keras.io/optimizers/`.