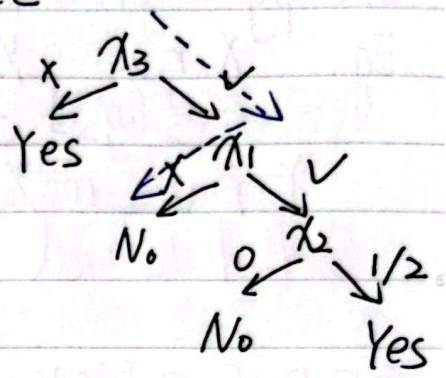
No.

Date

CS182: Introduction to Machine Learning

决策树 Decision Tree

x_1	x_2	x_3	y
✓	0	✓	No
✗	1	✓	No
✗	0	✗	Yes
✓	1	✓	Yes
✓	2	✗	Yes



想从 Feature x_1, x_2, x_3 判断 y，可用决策树建立流程

上表格用来做 training set，建右上图的树。则遇到：

$(x_1, x_2, x_3) = (✗, 2, ✓)$ 样例，决策流程如蓝笔所示

遇到 y 状态叶子结点，流程停止

那么，在树中，依据有先 x_3 后 x_1 后 x_2，能换顺序么？

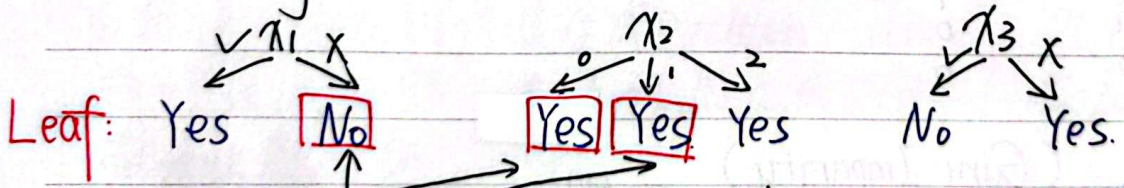
⇓

- Def: A splitting criterion is a function that measures how good or useful splitting on a particular feature is for a specified dataset.

Feature 顺序如何选？由 splitting criterion 量化

有以下常用标准：

① Training Error Rate:



判断 Leaf Node 为谁时：填啥 error 最低，就填啥；一样则均可

Error Rate: 4/5 3/5 4/5

则选 x_1 / x_3 均可

KOKUYO

② Mutual Information Gain

Entropy: $H(X) = -\sum_{v \in V(X)} P(X=v) \log_2 [P(X=v)]$

Eg: X r.v.: ½为0, ½为1. 则称X的熵为:

$-\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$

Y r.v.: 100%为0, $H(Y) = -1\log_2 1 = 0$

感性来说: 越不确定, 则熵越大

而又Def: $I(Y;X) = H(Y) - H(Y|X)$

$= H(Y) - \sum_{v \in V(X)} P(X=v)\, H(Y|X=v)$, i.e., $I(y; x_d)$

$= H(Y) - \sum_{v \in V(x_d)} f_v (H(Y_{x_d=v}))$

f_v → fraction of data points where $x_d = v$

$Y_{x_d=v}$ → set of all labels where $x_d = v$

Eg:

x_d	y
1	0
0	0

$I(y, x_d) = H(Y) - \sum_v H(Y_{x_d=v}) \cdot f_v$

$= 1 - \frac{1}{2} H(Y_{x_d=1}) - \frac{1}{2} H(Y_{x_d=0})$

$= 1$

x_d	y
1	1
0	0
0	0

$I(y, x_d) = H(Y) - \cdots$

$= 1 - \frac{1}{2} H(Y_{x_d=1}) - \frac{1}{2} H(Y_{x_d=0})$

$= 1 - \frac{1}{2}\cdot 1 - \frac{1}{2}\cdot 1 = 0$

则在:

x_1	x_2	y
1	0	1
0	0	0
0	0	0

这样的情况: 选 x_1 作为第一个决策点

③ 基尼指数 (Gini Impurity)

$\text{Gini}(D) = \sum_{k=1}^{|y|} \sum_{k' \neq k} p_k p_{k'} = 1 - \sum_{k=1}^{|y|} p_k^2$

$\text{Gini\_index} = \sum_{v=1}^{V} \frac{|D^v|}{|D|} \text{Gini}(D^v) \Rightarrow a^* = \underset{a \in A}{\text{argmin}}\ \text{Gini\_index}(D, a)$

No.

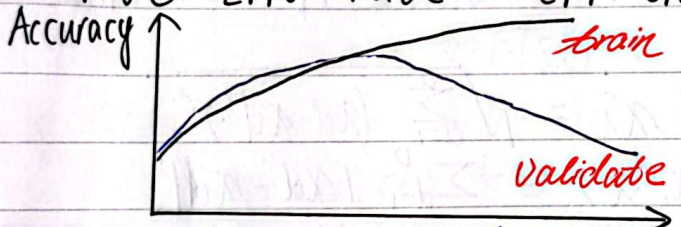
Date

在选出第一个feature之后，取 $x_i = v_i$ 的样本子集，抛开 x_i 因素，考虑选下一个 x_i

Decision Tree Pros: 可解释；高效；可用于分类/回归
但 Cons: splits consider immediate impact ; Overfit *

若 x_i 特征是 real-valued feature，则可以考虑划分分类区间，如：$(-\infty, a]$ 一类 $[a,b)$，$[b,+\infty)\cdots(b>a)$

*: Training Error = err(h, Dtrain)
Test Error = err(h, Dtest)
True Error rate = err(h)，h为所有样品



why overfitting ? 有时划分过程重复，branch过多
为了对付 Decision Tree 中的 overfitting，可采用剪枝处理：
分为 prepruning & post-pruning
预剪枝：对每个结点在划分前估计，若当前结点的划分不能带来泛化性能提升，则停止划分并当前结点标记为叶结点
后剪枝：先从训练集生成Decision Tree，然后自底向上对非叶结点考察，若该结点生成的子树换成叶子结点能带来性能升级，则换。
在 CS182 PPT中展示的仅为 post-pruning，并且顺序是自上而下。以西瓜书为主。
两策略(pre & post)均在讨论：一个非叶子结点是否有必要划分出新特征；pre是生成时便讨论 splitting feature (selected through criterion) 有必要么；post 是生成后自下而上讨论这一点。

KOKUYO