Chapter 4 Classification

AMA4680
Fu Guanxing
guanxing.fu@polyu.edu.hk

Classification

- Supervised learning.
- Let $\{x_i, y_i\}_{i=1}^n$ be a set of examples, where x_i's are from some set/space, and $y_i \in \{-1, 1\}$ are labels. Suppose the sample is drawn from some unknown yet fixed probability distribution, or in another way of saying, some existing and fixed pattern.[1]
- The classification task is to find some function f from the set where we draw x_i's, to the label set $\mathcal{Y} = \{-1, 1\}$,

$$f : \mathcal{X} \to \mathcal{Y},$$

  so that when some new datum x comes without label, we can have a large confidence that $f(x)$ is the right label that x should have. The function f is also called the ***classification model***.
- There are also "multi-class" classification tasks, where the label set $\{1, 2, \ldots, k\}$ has more than two elements.

[1] Each x_i can be a vector.

Outline

- Decision tree
- Naive Bayes classifier
- Support vector machine
- Logistic regression
- Artificial neural network

Decision trees

Decision tree

Usually, a decision tree includes one *root node*, several *internal nodes* and several *leaf nodes*:

- **Root node**: the first criterion. No incoming edge.
- **Internal node**: all the other criteria. Only one incoming edge, two or more outgoing edges.
- **Leaf/terminal node**: coresponding to decision; one incoming edge and no outgoing edge.

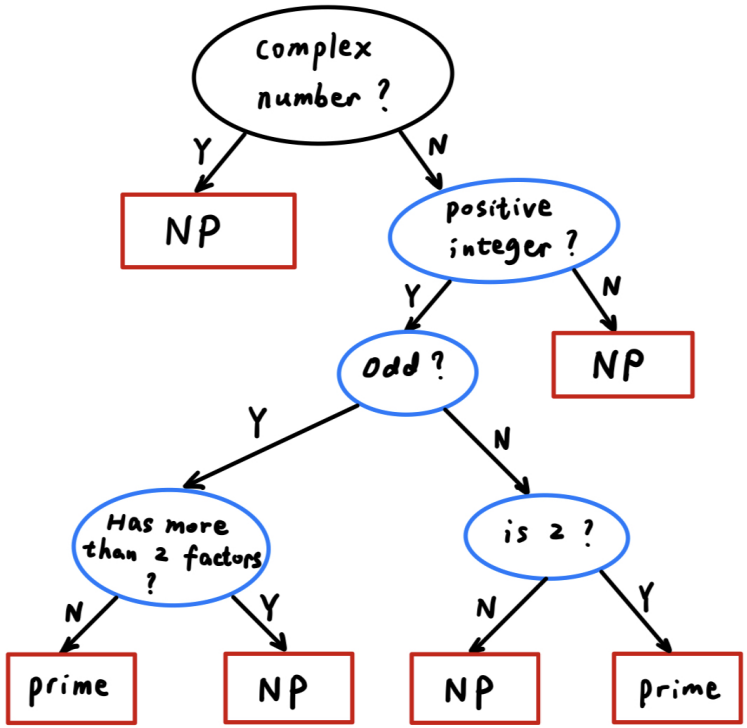
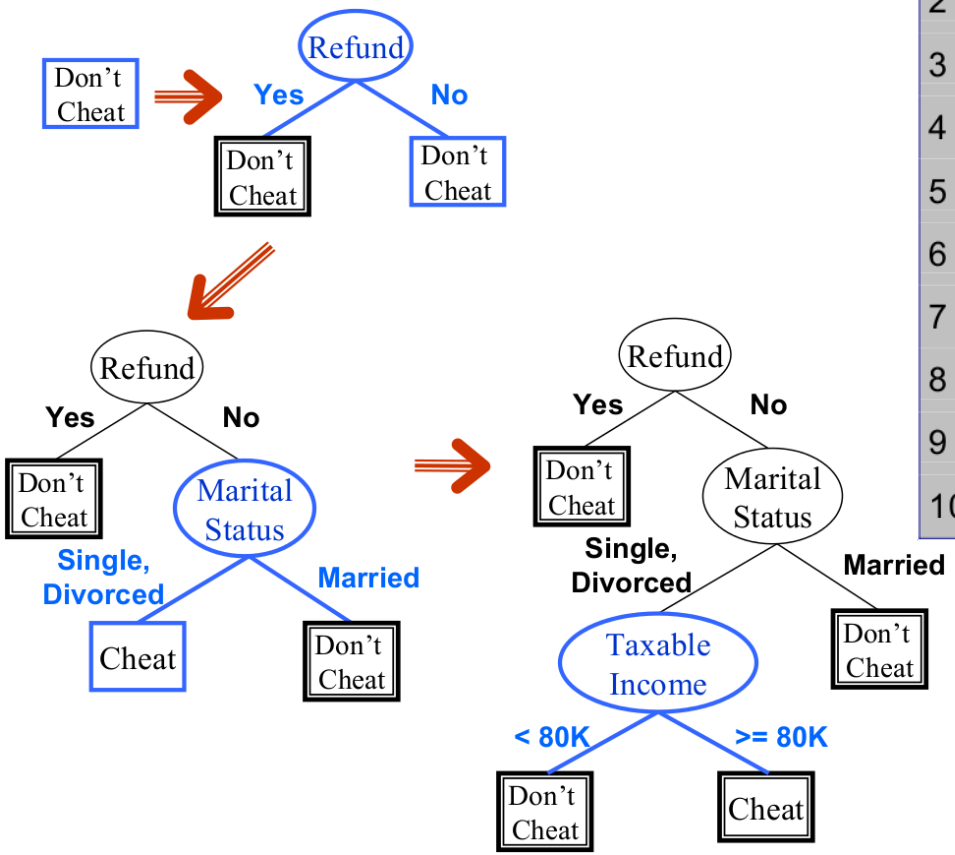


Figure: identification of prime numbers.

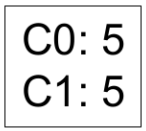
Example



Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

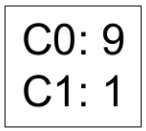
How to determine the best split?

- Which feature comes first?
- The more pure, the better.
- Consider the following splitting result by some criterion



Non-homogeneous,

High degree of impurity



Homogeneous,

Low degree of impurity

- Measures of node (im)purity:
  - Information entropy/gain.
  - Gini index.
  - Classification error.

Measures of impurity: information entropy (1/4)

- Let $\mathcal{D}$ be the dataset in node t.
- Let $p(i|t)$ denote the fraction of samples in $\mathcal{D}$ that belong to class $i \in \{1, \cdots, c\}$. Therefore,

$$\sum_{i=1}^{c} p(i|t) = 1, \text{ for any } t.$$

- The *information entropy* of $\mathcal{D}$ is defined as

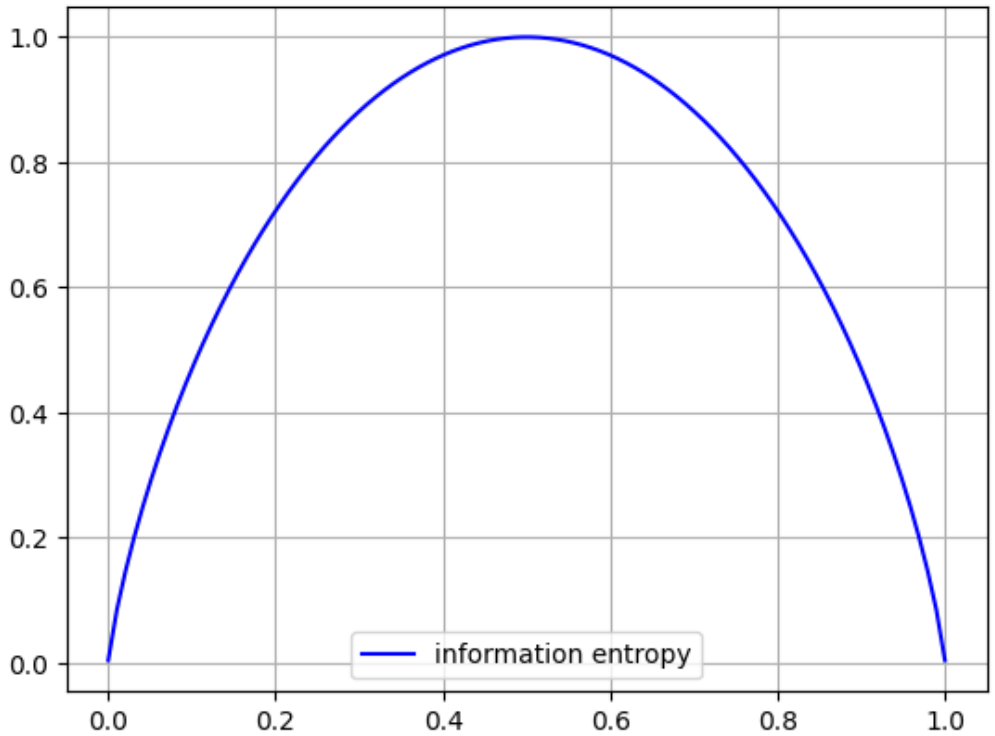
$$\text{Entropy}(\mathcal{D}) := -\sum_{i=1}^{c} p(i|t) \log_2 p(i|t).$$

Remark.

- $0 \times \log_2 0 := \lim_{x \to 0+} x \log_2 x = 0$.
- $\text{Entropy}(\mathcal{D}) \in [0, \log_2 c]$ (Exercise).
- The smaller $\text{Entropy}(\mathcal{D})$ is, the more pure $\mathcal{D}$ is.

Information entropy (2/4)

- Let $c = 2$ and write $p := p(1|t)$ and $p(2|t) = 1 - p$.



- $\arg\max_{p\in[0,1]} \text{Entropy}(p) = \frac{1}{2}$, and $\arg\min_{p\in[0,1]} \text{Entropy}(p) = \{0, 1\}$.

Information entropy (3/4)

- Recall the definition of entropy $-\sum_{i=1}^{c} p(i) \log_2 p(i)$.

node 1	count
class 1	0
class 2	6

  Entropy:

node 2	count
class 1	1
class 2	5

  Entropy:

The following two are left as exercises.

node 3	count
class 1	3
class 2	3

node 4	count
class 1	5
class 2	1

Information entropy (4/4)

- Assume feature[2] a has k options $\{v_1, \cdots, v_k\}$. $\mathcal{D}$ can be split into k groups:

$$\mathcal{D}^{v_i} = \{x \in \mathcal{D} : \text{feature } a \text{ of } x \text{ is } v_i\}, \qquad i = 1, \cdots, k.$$

- The *information gain* of $\mathcal{D}$ using feature a as the splitting criterion

$$\text{Gain}(\mathcal{D}, a) = \text{Entropy}(\mathcal{D}) - \sum_{j=1}^{k} \frac{|\mathcal{D}^{v_j}|}{|\mathcal{D}|} \text{Entropy}(\mathcal{D}^{v_j}),$$

- The best splitting criterion is $a^* = \arg\max_{a \in A} \text{Gain}(\mathcal{D}, a)$.
- **ID3** decision tree algorithm is based on information gain.

[2] If feature a is marital status, then $k = 3$.

Measure of impurity: Gini index (1/2)

- Recall $p(i|t) = \frac{\#\text{samples in } \mathcal{D} \text{ that belong to Class } i}{|\mathcal{D}|}$.
- Gini index of $\mathcal{D}$ is defined as

$$\text{Gini}(\mathcal{D}) = \sum_{i=1}^{c} \sum_{i' \neq i} p(i|t)p(i'|t) = 1 - \sum_{i=1}^{c} p(i|t)^2.$$

- Implication: Gini index of $\mathcal{D}$ is the probability that two randomly selected samples do not in the same class.
- Proof:

$$\sum_{i=1}^{c} \sum_{i' \neq i} p(i|t)p(i'|t) = \sum_{i=1}^{c} \left(\sum_{i'=1}^{c} p(i|t)p(i'|t) - p(i|t)^2 \right) = 1 - \sum_{i=1}^{c} p(i|t)^2.$$

- Let a be a feature with k options. The *Gini index* of $\mathcal{D}$ using feature a as the splitting criterion is defined as

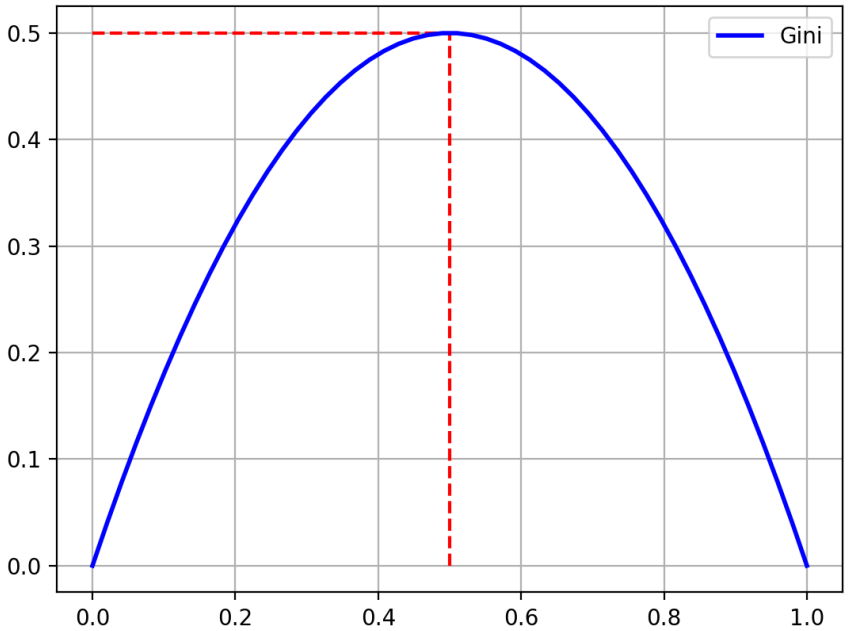
$$\text{Gini\_index}(\mathcal{D}, a) = \sum_{j=1}^{k} \frac{|\mathcal{D}^{v_j}|}{|\mathcal{D}|} \text{Gini}(\mathcal{D}^{v_j}).$$

- The splitting criterion for $\mathcal{D}$ is $a^* = \arg\min_{a \in A} \text{Gini\_index}(\mathcal{D}, a)$.
- **CART** algorithm is based on Gini index.

Gini index (2/2)

- The smaller Gini index is, the more pure the node is.
- Example: let $c = 2$ and write $p := p(1|t)$ and $p(2|t) = 1 - p$. Then

$$\text{Gini} = 1 - p^2 - (1-p)^2 = 2p - 2p^2.$$



- $p = \frac{1}{2}$
- $p = 0$ or 1

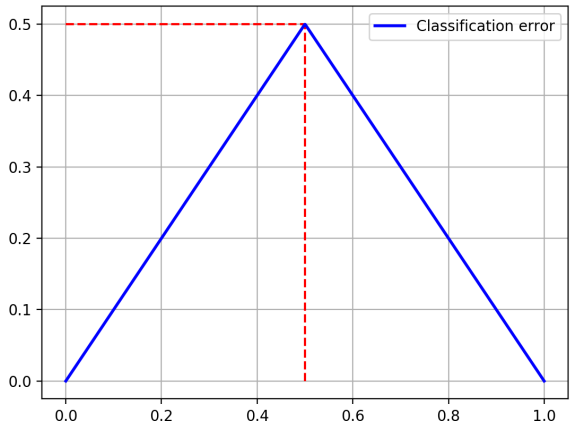
Measure of impurity: classification error

- Recall $p(i|t) = \frac{\#\text{samples in } \mathcal{D} \text{ that belong to Class } i}{|\mathcal{D}|}$.
- The classification error of $\mathcal{D}$ is

$$\text{ClassificationError}(\mathcal{D}) = 1 - \max_{i=1,\cdots,c} p(i|t)$$

- Example. Consider $c = 2$, $p := p(1|t)$ and $p(2|t) = 1 - p$. Then

$$\text{ClassificationError}(\mathcal{D}) = 1 - \max\{p, 1-p\}.$$



- The Classification Error of $\mathcal{D}$ using feature a as splitting criterion is

$$\text{ClassificationError}(\mathcal{D}, a) = \sum_{j=1}^{k} \frac{|\mathcal{D}^{v_j}|}{|\mathcal{D}|} \text{ClassificationError}(\mathcal{D}^{v_j}).$$

Example of decision tree: weather and soccer playing

Training Dataset

Index	Outlook	Temperature	Humidity	Wind	Play Soccer
1	Sunny	Mild	High	Weak	No
2	Rain	Mild	High	Strong	No
3	Rain	Cool	Normal	Strong	No
4	Sunny	Hot	High	Weak	No
5	Sunny	Hot	High	Strong	No
6	Overcast	Hot	High	Weak	Yes
7	Rain	Mild	High	Weak	Yes
8	Rain	Cool	Normal	Weak	Yes
9	Overcast	Cool	Normal	Strong	Yes
10	Sunny	Cool	Normal	Weak	Yes
11	Rain	Mild	Normal	Weak	Yes
12	Sunny	Mild	Normal	Strong	Yes
13	Overcast	Mild	High	Strong	Yes
14	Overcast	Hot	Normal	Weak	Yes

Construct a decision tree by using **information gain**.

Determination of the root node $\mathcal{D}_1$

Recall

$$\text{Gain}(\mathcal{D}, a) = \text{Entropy}(\mathcal{D}) - \sum_{j=1}^{k} \frac{|\mathcal{D}^{v_j}|}{|\mathcal{D}|} \text{Entropy}(\mathcal{D}^{v_j}),$$

where

$$\text{Entropy}(\mathcal{D}) := -\sum_{i=1}^{c} p(i|t) \log_2 p(i|t).$$

- Calculation of the information gain of **Outlook.** In $\mathcal{D}_1$, 5 samples are labeled *No* and 9 samples are labeled *Yes*. Thus,

$$\text{Entropy}(\mathcal{D}_1) = -\frac{5}{14}\log_2\frac{5}{14} - \frac{9}{14}\log_2\frac{9}{14} = 0.94$$

$$\text{Entropy}(\mathcal{D}_1, \mathbf{Outlook} = \text{Sun}) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971$$

$$\text{Entropy}(\mathcal{D}_1, \mathbf{Outlook} = \text{Over}) = -1\log_2 1 - 0\log_2 0 = 0$$

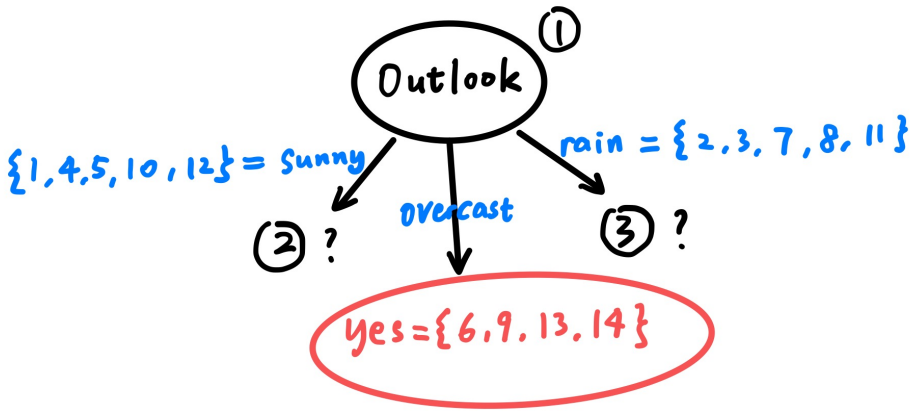
$$\text{Entropy}(\mathcal{D}_1, \mathbf{Outlook} = \text{Rain}) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971$$

Thus, $\text{Gain}(\mathcal{D}_1, \mathbf{Outlook}) = 0.94 - \frac{5}{14} \times 0.971 - \frac{4}{15} \times 0 - \frac{5}{14} \times 0.971 = 0.246$, which is larger than $\text{Gain}(\mathcal{D}_1, \mathbf{Tem})$, $\text{Gain}(\mathcal{D}_1, \mathbf{Hum})$ and $\text{Gain}(\mathcal{D}_1, \mathbf{Wind})$.

- The first splitting criterion is **Outlook** $\curvearrowright$.

Determination of internal node $\mathcal{D}_2$.

- $\mathcal{D}_2 = \{1, 4, 5, 10, 12\}$

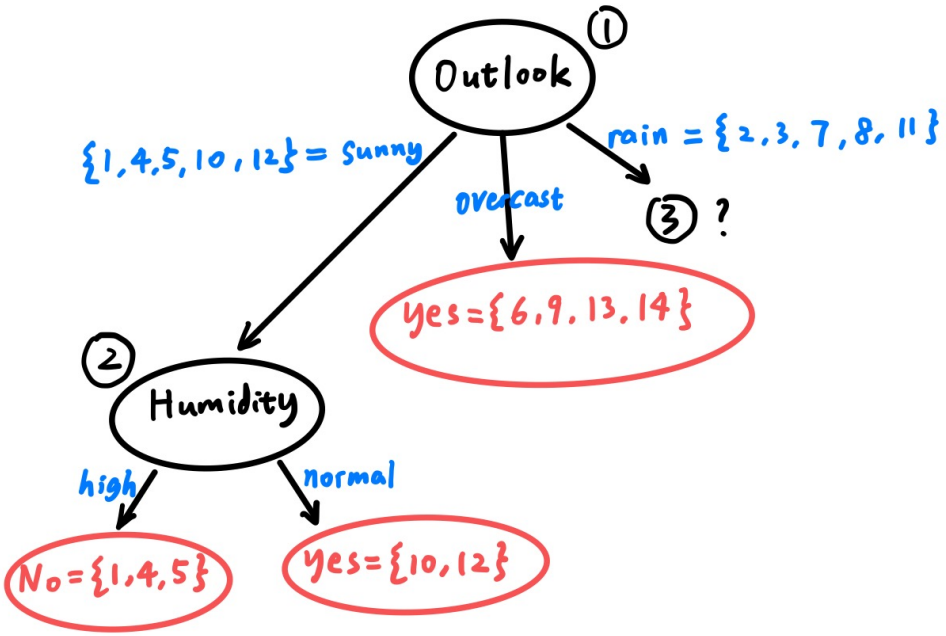


Index	Outlook	Temperature	Humidity	Wind	Play Soccer
1	Sunny	Mild	High	Weak	No
4	Sunny	Hot	High	Weak	No
5	Sunny	Hot	High	Strong	No
10	Sunny	Cool	Normal	Weak	Yes
12	Sunny	Mild	Normal	Strong	Yes

- Entropy$(\mathcal{D}_2) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971$ and Entropy$(\mathcal{D}_2, \mathbf{Hum} = \text{high}) =$ Entropy$(\mathcal{D}_2, \mathbf{Hum} = \text{normal}) = 0$. Thus, Gain$(\mathcal{D}_2, \mathbf{Hum}) = 0.971$, which is larger than Gain$(\mathcal{D}_2, \mathbf{Tem})$ and Gain$(\mathcal{D}_2, \mathbf{Wind})$ $\curvearrowright$

Determination of internal note $\mathcal{D}_3$

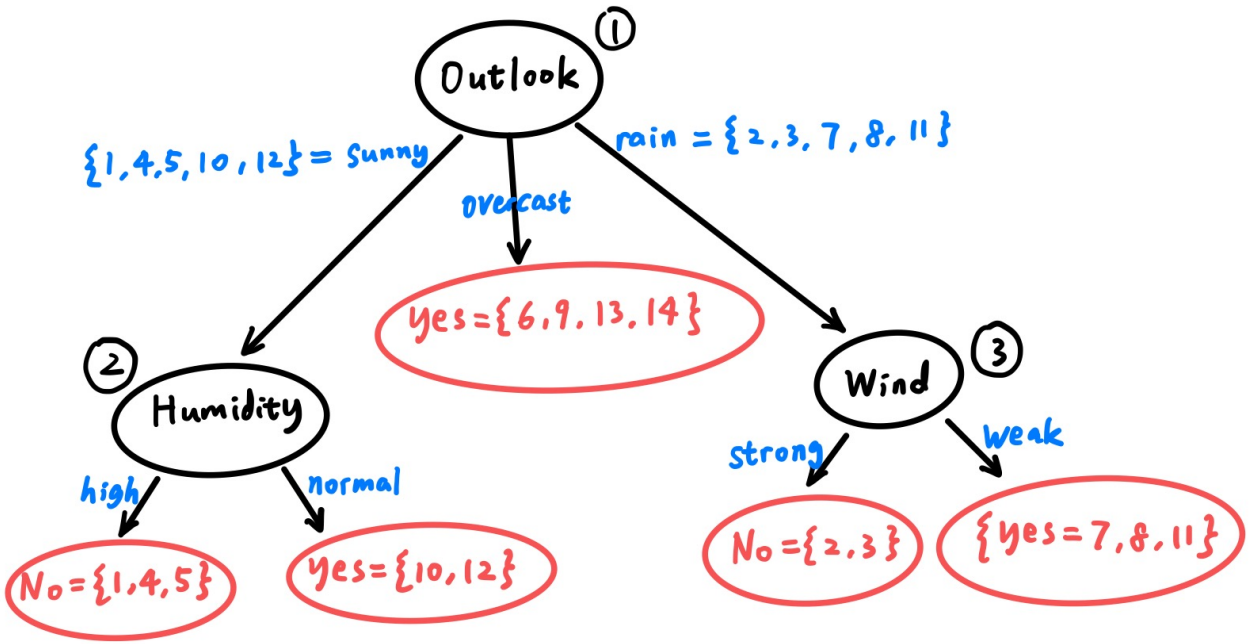
- $\mathcal{D}_3 = \{2, 3, 7, 8, 11\}$.



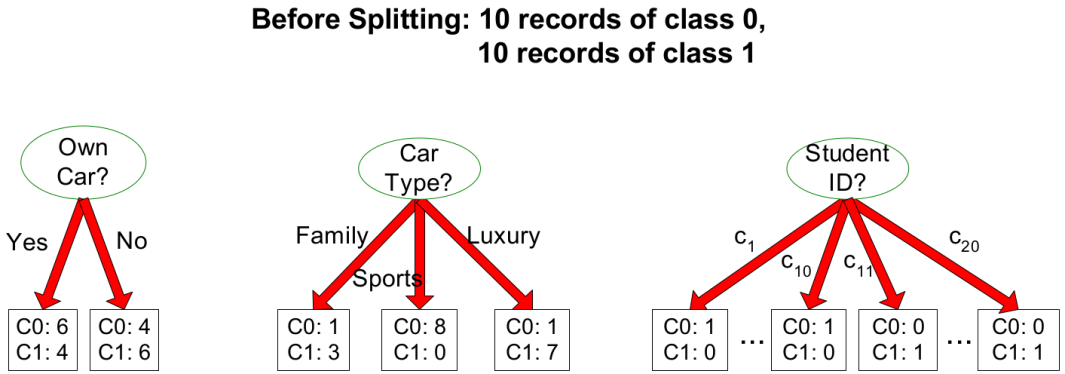
Index	Outlook	Temperature	Humidity	Wind	Play Soccer
2	Rain	Mild	High	Strong	No
3	Rain	Cool	Normal	Strong	No
7	Rain	Mild	High	Weak	Yes
8	Rain	Cool	Normal	Weak	Yes
11	Rain	Mild	Normal	Weak	Yes

- **Wind**.

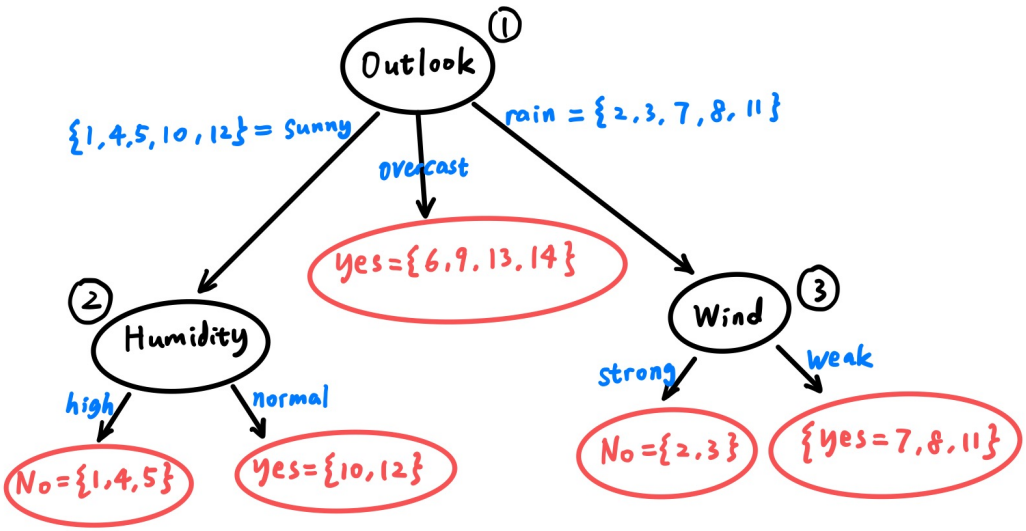
- Decision tree:



- End?



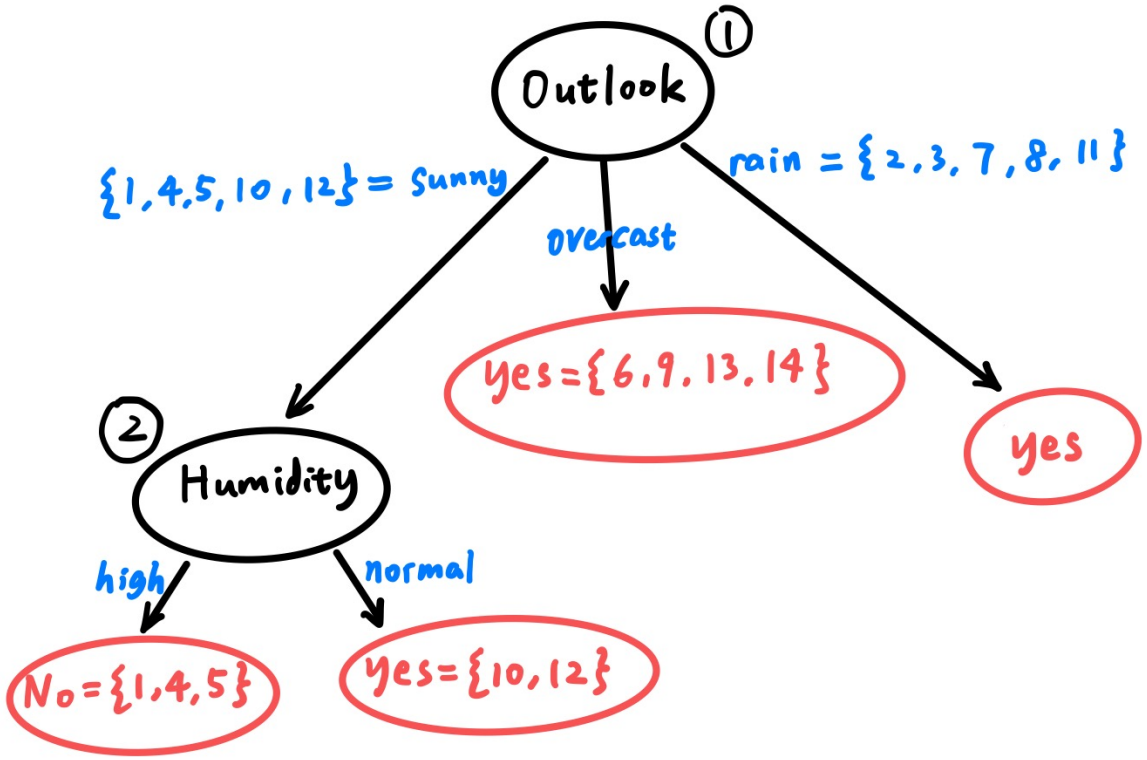
Post-pruning by validation data



Index	Outlook	Temperature	Humidity	Wind	Play Soccer
15	Sunny	Hot	Normal	Strong	Yes
16	Sunny	Mild	Normal	Weak	Yes
17	Overcast	Mold	Normal	Weak	No
18	Overcast	Cool	Normal	Strong	No
19	Rain	Cool	High	Strong	Yes

- The prediction rate is 40%.
- Cut Node 3 and change it to a leaf node $\Rightarrow$ the decision of Node 3 is *Yes* by vote $\Rightarrow$ Sample 19 is correctly predicted; prediction rate is increased to 60%.

Decision tree after pruning



Exercise

Consider the following data set for a binary class problem.

A	B	class label
T	F	+
T	T	+
T	T	+
T	F	−
T	T	+
F	F	−
F	F	−
F	F	−
T	T	−
T	F	−

(a). Calculate the entropy gain (information gain) when splitting on A and B. Which feature would the decision tree induction algorithm choose?

(b). Calculate the gain in the Gini index when splitting on A and B. Which feature would the decision tree induction algorithm choose?

Implication. Different impurity measures may suggest different attibutes for node-splitting.

Naive Bayes Classifier

Bayes rule

Recall conditional probability

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}; \quad \mathbb{P}(B|A) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(A)}.$$

Bayes' Theorem. Let $B_1, \cdots, B_n$ be a partition of the sample space S (i.e., for $i \neq j$, $B_i \cap B_j = \emptyset$ and $\cup_{k=1}^n B_k = S$), then for any $k = 1, \cdots n$ and any event A,

$$\mathbb{P}(B_k|A) = \frac{\mathbb{P}(B_k \cap A)}{\mathbb{P}(A)} = \frac{\mathbb{P}(B_k)\mathbb{P}(A|B_k)}{\mathbb{P}(B_1)\mathbb{P}(A|B_1) + \cdots + \mathbb{P}(B_n)\mathbb{P}(A|B_n)}.$$

Bayes classifier (1/3)

- Let C denote the r.v. of label and A be the r.v. of features[3].
- Let $\mathcal{Y} = \{c_1, \cdots, c_m\}$ be the label space.
- Let $\boldsymbol{x} := (a_1, \cdots, a_n)$ be an observation of features. Which class does it belong to:

$$f^*(a_1, \cdots, a_n) = \arg\max_{c \in \mathcal{Y}} \mathbb{P}\left(C = c | A = (a_1, \cdots, a_n)\right).$$

Example. Let $\mathcal{Y} = \{\text{yes}, \text{no}\}$.

Remark. For each $c \in \mathcal{Y}$ and $x = (a_1, \cdots, a_n)$

  - $\mathbb{P}(c|\boldsymbol{x}) := \mathbb{P}(C = c | A = \boldsymbol{x})$

- Bayes Classifiers:
  - $\mathbb{P}(c|a_1, a_2, \ldots, a_n) = \frac{\mathbb{P}(a_1, a_2, \ldots, a_n|c)\mathbb{P}(c)}{\mathbb{P}(a_1, a_2, \ldots, a_n)}$
  - $\arg\max_{c \in \mathcal{Y}} \mathbb{P}(c|a_1, a_2, \ldots, a_n) = \arg\max_{c \in \mathcal{Y}} \mathbb{P}(a_1, a_2, \ldots, a_n|c)\mathbb{P}(c)$
  - Estimate two probabilities $\curvearrowright$

[3] $A = (A_1, \cdots, A_n)$

Bayes classifier (2/3)

- $\mathbb{P}(c) \approx \frac{\#\{\text{samples in the training set that belong to Class } c\}}{\#\{\text{training data}\}}$
- In discrete-valued features, $(a_1, \cdots, a_n)$ is one value of features.

$$\mathbb{P}(a_1, \cdots, a_n|c)$$
$$\approx \frac{\#\{\text{samples in training data that belong to Class } c \text{ with feature } (a_1, \cdots, a_n)\}}{\#\{\text{training data that belong to Class } c\}}$$

- In continuous-valued features, $(a_1, \cdots, a_n)$ is a domain of features.

$$\mathbb{P}(a_1, \cdots, a_n|c) = \mathbb{P}(A \in (a_1, \cdots, a_n)|c)$$
$$\approx \frac{\#\{\text{samples in training data that belong to Class } c \text{ with feature } \in (a_1, \cdots, a_n)\}}{\#\{\text{training data that belong to Class } c\}}$$

- In mixed feature, some a_i are values and some are domains.

$$\mathbb{P}(a_1, \cdots, a_n|c) = \mathbb{P}(A \in (a_1, \cdots, a_n)|c)$$
$$\approx \frac{\#\{\text{samples in training data that belong to Class } c,\ A_1 = a_1, A_2 \in a_2, \cdots\}}{\#\{\text{training data that belong to Class } c\}}$$

Bayes classifier (3/3)

- Recall

$$\mathbb{P}(a_1, \cdots, a_n|c)$$
$$\approx \frac{\#\{\text{samples in training data that belong to Class } c \text{ with feature } (a_1, \cdots, a_n)\}}{\#\{\text{training data that belong to Class } c\}}$$

- **Disadvantage:** Too many possible combinations of $(a_1, \cdots, a_n)$ (2^n if each feature has two options). In reality, 2^n is much larger than (training) sample size, such that some feature combination $(a_1^0, \cdots, a_n^0)$ may not appear in training samples!

- $\curvearrowright$ Naive Bayes classifier!

Naive Bayes classifier (1/2)

- **Attribute conditional independence assumption**: for any class c and any attributes $(a_1, \cdots, a_n)$,

$$\mathbb{P}(a_1, a_2, \ldots, a_n | c) = \prod_{i=1}^{n} \mathbb{P}(a_i | c).$$

  - Each feature/attribute influences the classification independently.
  - **Advantage.** One can estimate $\mathbb{P}(a_i|c)$ for each a_i and c; the estimate is not influenced by the large combination.
- $(a_1, \cdots, a_n)$ is classified to c^*, which is the maximizer of

$$\mathbb{P}(a_1, a_2, \ldots, a_n | c)\mathbb{P}(c) = \mathbb{P}(c) \prod_{i=1}^{n} \mathbb{P}(a_i | c),$$

where

$$\mathbb{P}(a_i|c) \approx \frac{\#\{\text{training samples in Class } c \text{ with i}-\text{th feature } a_i\}}{\#\{\text{training samples in Class } c\}}.$$

- **Disadvantage:** ↷

Naive Bayes classifier (2/2)

- **Disadvantage.** If training samples in Class c do not contain any sample with $A_i = a_i$, then $\mathbb{P}(a_i|c) = 0$, which implies that

$$\mathbb{P}(a_1, \cdots, a_i, \cdots, a_n|c) = 0 \Rightarrow \mathbb{P}(c|a_1, \cdots, a_i, \cdots, a_n) = 0.$$

- **Remedy.** Recall the original probability:

$$\mathbb{P}(a_i|c) = \frac{N_{ic}}{N_c}, \qquad \mathbb{P}(c) = \frac{N_c}{N}.$$

Laplacian smoothing:

$$\mathbb{P}(a_i|c) = \frac{N_{ic} + \alpha}{N_c + \alpha \times (\#A_i)}, \qquad \mathbb{P}(c) = \frac{N_c + \alpha}{N + \alpha \times (\#C)},$$

where

- $(\#C)$ denotes the number of all possible classes in the training sample.
- $(\#A_i)$ denotes the number of options feature i can choose; e.g., $A_i =$ "smoke or not", then $(\#A_i) = 2$.
- $\mathbb{P}(\text{smoke}|c) + \mathbb{P}(\text{not}|c) = \frac{N_{\text{smoke},c}+\alpha}{N_c+2\alpha} + \frac{N_{\text{not},c}+\alpha}{N_c+2\alpha} = 1.$
- $\alpha > 0$ is a parameter. In this subject, we will always set $\boxed{\alpha = 1}$.

Example: Naive Bayes classifier with Laplacian smoothing

Training sample:

Name	Give Birth	Lay Eggs	Can Fly	Have Legs	Class
monkey	yes	no	no	yes	mammal
whale	yes	no	no	no	mammal
bat	yes	no	yes	yes	mammal
cat	yes	no	no	yes	mammal
dolphin	yes	no	no	no	mammal
python	no	yes	no	no	non-mammal
salmon	no	yes	no	no	non-mammal
frog	no	yes	no	yes	non-mammal
lizard	no	yes	no	yes	non-mammal
pigeon	no	yes	yes	yes	non-mammal
leopard shark	yes	no	no	no	non-mammal
turtle	no	yes	no	yes	non-mammal
penguin	no	yes	no	yes	non-mammal
owl	no	yes	yes	yes	non-mammal
eagle	no	yes	yes	yes	non-mammal

Example, *cont'd*

New sample:

Name	Give Birth	Lay Eggs	Can Fly	Have Legs	Class
human	yes	no	no	yes	?

Name	Give Birth	Lay Eggs	Can Fly	Have Legs	Class
platypus	no	yes	no	yes	?

Conclusion.

- The prediction is *non-mammal*. However, the true class is *mammal*.

Reason.

- Not enough training sample.
- platypus only appears in Australia; less similarity with other mammals.