Decision Trees

Ninghao Liu

University of Georgia

February 6, 2024

Some contents adopted from “Data Mining”, Section 8.2, by Jiawei Han et al.

Decision Trees Induction

Basic algorithm:

- At start, all the training examples are at the root.
- Attributes are **categorical** (if continuous-valued, they are discretized in advance).
- Examples are **partitioned recursively** based on selected attributes.
- Test attributes are **selected** on the basis of a heuristic or statistical measure (e.g., information gain).

Conditions for stopping:

- All samples for a given node belong to the same class.
- There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf.
- There are no samples left.

Decision Trees Induction

Brief history of decision tree learning:

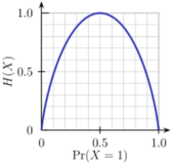
- **ID3**.
- **C4.5**.
- **CART**.

How to Select Attributes?

- Entropy (Information Theory)
  - A measure of uncertainty associated with a random variable
  - Calculation: For a discrete random variable Y taking m distinct values $\{y_1, \ldots, y_m\}$,
    - $H(Y) = -\sum_{i=1}^{m} p_i \log(p_i)$, where $p_i = P(Y = y_i)$

How to Select Attributes?

- Entropy (Information Theory)
  - A measure of uncertainty associated with a random variable
  - Calculation: For a discrete random variable Y taking m distinct values $\{y_1, \dots, y_m\}$,
    - $H(Y) = -\sum_{i=1}^{m} p_i \log(p_i)$, where $p_i = P(Y = y_i)$
  - Interpretation:
    - Higher entropy => higher uncertainty
    - Lower entropy => lower uncertainty



m = 2

How to Select Attributes?

- Example 1: The random variable Y: Whether the sun will rise tomorrow. Can you compute the entropy of Y?
- Example 2: The random variable Y: The winning number sequence of powerball. Can you compute the entropy of Y?
  - 4 balls.
  - The number on each ball ranges from 1 to 8.

How to Select Attributes?

- Example 3: The random variable Y: Whether the sun will rise tomorrow. I am telling you that "the sun will rise tomorrow".
- Example 4: The random variable Y: The winning number sequence of powerball.
  - 4 balls.
  - The number on each ball ranges from 1 to 8.
  - I am telling you: "The winning numbers tomorrow are [1, 3, 7, A], where A is a number from $\{4, 5, 6, 7\}$".
  - Does the sentence above contain much information or little information?

Information Gain

The expected **information** needed to classify instances in $\mathcal{D}$ is given by:

$$Info(\mathcal{D}) = -\sum_{i=1}^{m} p_i \log_2(p_i) \tag{1}$$

where

- p_i is the nonzero probability that an arbitrary instance belong to class C_i
- $p_i = C_{i,\mathcal{D}}/|\mathcal{D}|$, and $C_{i,D}$ denotes the number of C_i instances in $\mathcal{D}$.

Information Gain

- We were to partition the instances in $\mathcal{D}$ on some attribute A having v distinct values $\{a_1, a_2, ..., a_v\}$.
- The above attribute will partition $\mathcal{D}$ into $\{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_v\}$.
- $\mathcal{D}_j$ contains instances whose value of A is a_j.
- These partitions correspond to the branches in the decision tree.
- Ideally, we would like this partitioning to produce an exact classification of the instances.
    - This is not easy.
    - A partition $\mathcal{D}_j$ may contain a collection of instances from different classes rather than from a single class.

Information Gain

- How much more information would we still need (after the partitioning) to arrive at an exact classification?
- This amount of information is measured by:

$$Info_A(\mathcal{D}) = \sum_{j}^{v} \frac{|\mathcal{D}_j|}{|\mathcal{D}|} \times Info(\mathcal{D}_j) \tag{2}$$

  - $\frac{|\mathcal{D}_j|}{|\mathcal{D}|}$ is the weight of the j-th partition.
  - $Info_A(\mathcal{D})$ is smaller $\rightarrow$ the greater the purity of the partitions.

Information Gain

$$Info(\mathcal{D}) = -\sum_{i=1}^{m} p_i \log_2(p_i) \tag{3}$$

$$Info_A(\mathcal{D}) = \sum_{j=1}^{v} \frac{|\mathcal{D}_j|}{|\mathcal{D}|} \times Info(\mathcal{D}_j) \tag{4}$$

- **Information gain**:

$$Gain(A) = Info(\mathcal{D}) - Info_A(\mathcal{D}) \tag{5}$$

  - The difference between the original information requirement and the new requirement (after partitioning on A).
  - The attribute A with the **highest information gain**, $Gain(A)$, is chosen as the splitting attribute.
  - We want to partition on the attribute A that would do the "best classification".

Decision Tree Induction

age	income	student	credit_rating	com
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Decision Tree Induction

- Class P: buys_computer = “yes”
- Class N: buys_computer = “no”

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Decision Tree Induction

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Decision Tree Induction

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$$Info_{age}(D) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2) = 0.694$$

$\frac{5}{14}I(2,3)$ means "age <=30" has 5 out of 14 samples, with 2 yes'es and 3 no's. Hence

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

Decision Tree Induction

- Class P: buys_computer = "yes"
- Class N: buys_computer = "no"

$$Info(D) = I(9,5) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940$$

age	p_i	n_i	$I(p_i, n_i)$
<=30	2	3	0.971
31…40	4	0	0
>40	3	2	0.971

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

$$Info_{age}(D) = \frac{5}{14}I(2,3) + \frac{4}{14}I(4,0) + \frac{5}{14}I(3,2) = 0.694$$

$\frac{5}{14}I(2,3)$ means "age <=30" has 5 out of 14 samples, with 2 yes'es and 3 no's. Hence

$$Gain(age) = Info(D) - Info_{age}(D) = 0.246$$

Similarly,

$$Gain(income) = 0.029$$
$$Gain(student) = 0.151$$
$$Gain(credit\_rating) = 0.048$$

Decision Tree Induction

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
<=30	medium	yes	excellent	yes

Decision Tree Induction

Information gain is not perfect. A notable problem occurs when information gain is applied to **attributes that can take on a large number of distinct values**.

- Suppose that one is building a decision tree for some data describing the customers of a business
- One of the input attributes might be the customer's membership number, if they are a member of the business's membership program.
- This attribute has a high mutual information, because it uniquely identifies each customer, but we do not want to include it in the decision tree.
- Deciding how to treat a customer based on their membership number is unlikely to generalize to customers we haven't seen before (**overfitting**).

Gain Ratio for Attribute Selection (C4.5)

- C4.5 (a successor of ID3) uses gain ratio to overcome the problem (normalization to information gain)

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2(\frac{|D_j|}{|D|})$$

  - GainRatio(A) = Gain(A)/SplitInfo(A)
- Ex. $SplitInfo_{income}(D) = -\frac{4}{14} \times \log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times \log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times \log_2\left(\frac{4}{14}\right) = 1.557$
  - gain_ratio(income) = 0.029/1.557 = 0.019
- The attribute with the maximum gain ratio is selected as the splitting attribute

Gain Ratio for Attribute Selection (C4.5)

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

For the example on the whiteboard:

- What is the *Gain* value for choose each of the attributes?
- Which attribute will be chosen using *GainRatio*?

Gini Index (CART)

- If a data set D contains examples from m classes, gini index, $gini(D)$ is defined as

$$
\begin{aligned}
gini(D) &= \sum_{i=1}^{m} \sum_{i' \neq i} p_i p_{i'} \\
&= 1 - \sum_{i=1}^{m} p_i^2
\end{aligned} \tag{6}
$$

  where p_i is the probability that an instance belongs to the i-th class.

- The gini index measures the "impurity" of a dataset, i.e., how likely that two instances in the dataset do not belong to the same class.

Gini Index (CART)

- If a data set D is split on A into two subsets D_1 and D_2, the *gini* index *gini*(D) is defined as

$$gini_A(D) = \frac{|D_1|}{|D|} gini(D_1) + \frac{|D_2|}{|D|} gini(D_2)$$

- Reduction in Impurity:

$$\Delta gini(A) = gini(D) - gini_A(D)$$

- The attribute provides the smallest $gini_{split}(D)$ (or the largest reduction in impurity) is chosen to split the node (*need to enumerate all the possible splitting points for each attribute*)

Gini Index (CART)

- Ex. D has 9 tuples in buys_computer = "yes" and 5 in "no"

$$gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

- Suppose the attribute income partitions D into 10 in D_1: {low, medium} and 4 in D_2

$$gini_{income \in \{low, medium\}}(D) = \left(\frac{10}{14}\right) Gini(D_1) + \left(\frac{4}{14}\right) Gini(D_2)$$

$$\begin{aligned} &= \frac{10}{14}\left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right) \\ &= 0.443 \\ &= Gini_{income \in \{high\}}(D). \end{aligned}$$

$Gini_{\{low,high\}}$ is 0.458; $Gini_{\{medium,high\}}$ is 0.450. Thus, split on the {low,medium} (and {high}) since it has the lowest Gini index

- All attributes are assumed continuous-valued
- May need other tools, e.g., clustering, to get the possible split values
- Can be modified for categorical attributes

Comparing Attribute Selection Measures

- The three measures, in general, return good results but
  - **Information gain**:
    - biased towards multivalued attributes
  - **Gain ratio**:
    - tends to prefer unbalanced splits in which one partition is much smaller than the others
  - **Gini index**:
    - biased to multivalued attributes
    - has difficulty when # of classes is large
    - tends to favor tests that result in equal-sized partitions and purity in both partitions