

Decision Trees

COMP9417, 23T1

1 Decision Trees

2 1 (a, b, c)

3 2 (a, b)

4 3

Section 1

Decision Trees

Decision Trees

A tree-like model used for both **regression** and **classification**.

Decision Trees

A tree-like model used for both **regression** and **classification**.

Advantages:

Decision Trees

A tree-like model used for both **regression** and **classification**.

Advantages:

- Interpretable
- Useful when used in ensemble learning (we'll come back to this notion)

Decision Trees

A tree-like model used for both **regression** and **classification**.

Advantages:

- Interpretable
- Useful when used in ensemble learning (we'll come back to this notion)

Disadvantages:

Decision Trees

A tree-like model used for both **regression** and **classification**.

Advantages:

- Interpretable
- Useful when used in ensemble learning (we'll come back to this notion)

Disadvantages:

- Tend to overfit data
- Often inaccurate in their most basic form

Entropy

Entropy essentially measures the *uncertainty* or *surprise* of a random variable.

We define the entropy for a set S ,

$$H(S) = \sum_{x \in X} -p(x) \log p(x)$$

where $p(x)$ represents the *proportion* of x in S .

Entropy

Entropy essentially measures the *uncertainty* or *surprise* of a random variable.

We define the entropy for a set S ,

$$H(S) = \sum_{x \in X} -p(x) \log p(x)$$

where $p(x)$ represents the *proportion* of x in S .

Say we have a random variable $X \sim \text{Bernoulli}(p)$. We can define the entropy of X :

Entropy

Entropy essentially measures the *uncertainty* or *surprise* of a random variable.

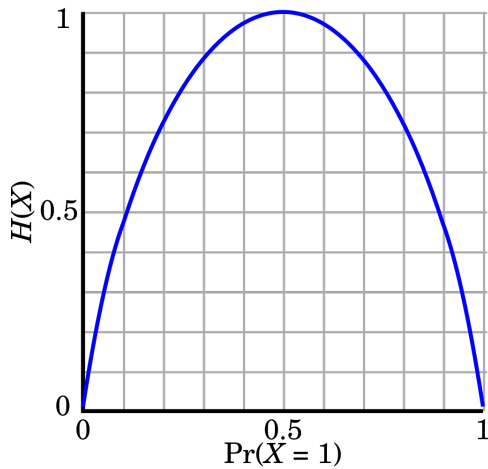
We define the entropy for a set S ,

$$H(S) = \sum_{x \in X} -p(x) \log p(x)$$

where $p(x)$ represents the *proportion* of x in S .

Say we have a random variable $X \sim \text{Bernoulli}(p)$. We can define the entropy of X :

$$H(x) = -(1-p) \log(1-p) - p \log p$$



Gain

To measure the *information* we gain by splitting on an attribute A for a dataset S , we define:

Gain

To measure the *information* we gain by splitting on an attribute A for a dataset S , we define:

$$\text{Gain}(S, A) = \text{Current entropy} - \text{Entropy if we split on } A$$

Gain

To measure the *information* we gain by splitting on an attribute A for a dataset S , we define:

$$\text{Gain}(S, A) = \text{Current entropy} - \text{Entropy if we split on } A$$

If we have a dataset S with a feature A ,

$$\text{Gain}(S, A) = H(S) - \sum_{v \in V_A} \frac{|S_v|}{|S|} H(S_v)$$


Section 2

1 (a, b, c)

1 (a, b, c)

Give decision trees to represent the following Boolean functions, where the variables A, B, C and D have values t or f , and the class value is either True or False. Can you observe any effect of the increasing complexity of the functions on the form of their expression as decision trees?

1a

 $A \wedge \neg B$

1a

Ⓐ $A \wedge \neg B$

A = t:

| B = f: True

| B = t: False

A = f: False

1b

ⓑ $A \vee [B \wedge C]$

1b

ⓑ $A \vee [B \wedge C]$

A = t: True

A = f:


| B = f: False

| B = t:

| C = t: True

| C = f: False

1c

 $A \text{ XOR } B$

1c

Ⓢ $A \text{ XOR } B$

$A = t$:

| $B = t$: False

| $B = f$: True

$A = f$:

| $B = t$: True

| $B = f$: False

1d

Ⓒ $[A \wedge B] \vee [C \wedge D]$

1d

$$\textcircled{c} \quad [A \wedge B] \vee [C \wedge D]$$

A = t:

| B = t: True

| B = f:

| C = t:

| D = t: True

| D = f: False

| C = f: False

A = f:

| C = t:

| D = t: True

| D = f: False

| C = f: False

Section 3

2 (a, b)

2a

Assume we learn a decision tree to predict class Y given attributes A , B and C from the following training set, with no pruning.

What would be the training error for this dataset?

A	B	C	Y
0	0	0	0
0	0	1	0
0	0	1	0
0	1	0	0
0	1	1	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	0	1
1	1	1	0
1	1	1	1

2a

Assume we learn a decision tree to predict class Y given attributes A , B and C from the following training set, with no pruning.

What would be the training error for this dataset?

We can shortcut this process, the attribute combinations $(0, 1, 1)$ and $(1, 1, 1)$ appear in both classes, therefore we will make an error on 2 points. So, our error is $2/12$ or $1/6$.

A	B	C	Y
0	0	0	0
0	0	1	0
0	0	1	0
0	1	0	0
0	1	1	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	0	1
1	1	1	0
1	1	1	1

2b

One nice feature of decision tree learners is that they can learn trees to do multi-class classification, i.e., where the problem is to learn to classify each instance into exactly one of $k > 2$ classes. Suppose a decision tree is to be learned on an arbitrary set of data where each instance has a discrete class value in one of $k > 2$ classes. What is the maximum training set error, expressed as fraction, that any dataset could have?

2b

One nice feature of decision tree learners is that they can learn trees to do multi-class classification, i.e., where the problem is to learn to classify each instance into exactly one of $k > 2$ classes. Suppose a decision tree is to be learned on an arbitrary set of data where each instance has a discrete class value in one of $k > 2$ classes. What is the maximum training set error, expressed as fraction, that any dataset could have?

If we have k classes and k points - in the worst case we have a sample for each class. Then we can only classify one point correctly in the entire dataset so our error is:

$$1 - \frac{1}{k} = \frac{k-1}{k}$$

Where we essentially rewrite the problem in terms of fractions of the training set.

Section 4

3

3

Look at the examples. Can you guess which attribute(s) will be most predictive of the class?

species	rebel	age	ability	homeworld
pearl	yes	6000	regeneration	no
bismuth	yes	8000	regeneration	no
pearl	no	6000	weapon-summoning	no
garnet	yes	5000	regeneration	no
amethyst	no	6000	shapeshifting	no
amethyst	yes	5000	shapeshifting	no
garnet	yes	6000	weapon-summoning	no
diamond	no	6000	regeneration	yes
diamond	no	8000	regeneration	yes
amethyst	no	5000	shapeshifting	yes
pearl	no	8000	shapeshifting	yes
jasper	no	6000	weapon-summoning	yes

You probably guessed that attributes 3 and 4 were not very predictive of the class, which is true. However, you might be surprised to learn that attribute “species” has higher information gain than attribute “rebel”. Why is this?

You probably guessed that attributes 3 and 4 were not very predictive of the class, which is true. However, you might be surprised to learn that attribute “species” has higher information gain than attribute “rebel”. Why is this?

Suppose you are told the following: for attribute “species” the Information Gain is 0.52 and Split Information is 2.46, whereas for attribute “rebel” the Information Gain is 0.48 and Split Information is 0.98.

You probably guessed that attributes 3 and 4 were not very predictive of the class, which is true. However, you might be surprised to learn that attribute “species” has higher information gain than attribute “rebel”. Why is this?

Suppose you are told the following: for attribute “species” the Information Gain is 0.52 and Split Information is 2.46, whereas for attribute “rebel” the Information Gain is 0.48 and Split Information is 0.98.

Which attribute would the decision-tree learning algorithm select as the split when using the Gain Ratio criterion instead of Information Gain? Is Gain Ratio a better criterion than Information Gain in this case?

We are given:

- $\text{Gain}(\text{species}) = 0.52$ and $\text{SI}(\text{species}) = 2.46$
- $\text{Gain}(\text{rebel}) = 0.48$ and $\text{SI}(\text{rebel}) = 0.98$

We are given:

- $\text{Gain}(\text{species}) = 0.52$ and $\text{SI}(\text{species}) = 2.46$
- $\text{Gain}(\text{rebel}) = 0.48$ and $\text{SI}(\text{rebel}) = 0.98$

The formula for Gain Ratio is just:

$$\text{GainRatio}(V) = \frac{\text{Gain}(V)}{\text{SI}(V)}$$

so, we have $\text{GainRatio}(\text{species}) = 0.21$ and $\text{GainRatio}(\text{rebel}) = 0.49$.

We are given:

- $\text{Gain}(\text{species}) = 0.52$ and $\text{SI}(\text{species}) = 2.46$
- $\text{Gain}(\text{rebel}) = 0.48$ and $\text{SI}(\text{rebel}) = 0.98$

The formula for Gain Ratio is just:

$$\text{GainRatio}(V) = \frac{\text{Gain}(V)}{\text{SI}(V)}$$

so, we have $\text{GainRatio}(\text{species}) = 0.21$ and $\text{GainRatio}(\text{rebel}) = 0.49$.

Therefore we pick **rebel** in this case as it has a higher gain ratio. It is a better choice as *species* only has high gain due to the number of values it takes.