Machine learning
Symbolic techniques

Russell and Norvig, Chapters 18, 19

Machine learning: Symbolic techniques
Overview: aims

- know of several symbolic machine learning techniques and representations
- decision tree learning
- current-best learning
- using decision trees and logic rules as representations respectively

Machine learning: Symbolic techniques
Overview: topics

- Decision tree learning
- Current-best-learning

Decision tree (1)

- Symbolic representation of acquired knowledge
- One of the simplest, most successful forms of learning algorithm
- Construction: input of a set of examples that are made up of attributes and an output
- Prediction: input of a situation made up of attributes, the tree returns the predicted output value

Decision tree (2)

- Decision made by performing a sequence of tests on the attributes
- Each internal node is a test
- Each branch is the possible result of the test
- Each leaf node specifies the value returned if reached

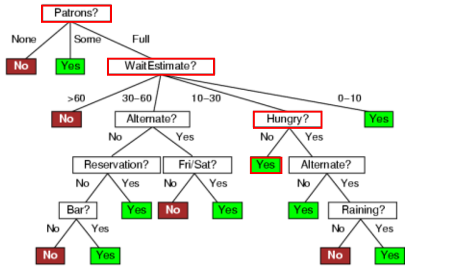
Decision tree (3)

- Inputs and outputs can be discrete or continuous (classification or regression)
- Will consider Boolean classification in more detail

Decision tree example: Wait for a table? (1)

- Goal predicate: WillWait
- Attributes:
  - Alternate
  - Bar
  - Fri/Sat
  - Hungry
  - Patrons
  - Price
  - Raining
  - Reservation
  - Type
  - WaitEstimate

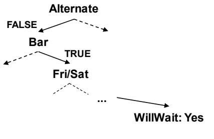
Decision tree example: Wait for a table? (2)



∀r Patrons(r, Full) ∧ WaitEstimate(r, "10-30") ∧ Hungry(r, No) ⇒ WillWait(r)

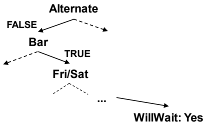
Inducing a decision tree from examples: How? (1)

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE
FALSE	TRUE	TRUE	FALSE	Full	$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$$$	FALSE	TRUE	Italian	10-30	FALSE
FALSE	FALSE	FALSE	FALSE	None	$	FALSE	FALSE	Thai	0-10	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$	FALSE	FALSE	Burger	30-60	TRUE



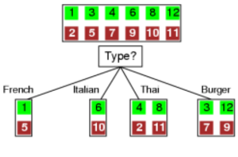
Inducing a decision tree from examples: How? (2)

- Large trees just "memorise" all cases
- Use every attribute of every example
- Large trees do not extrapolate to unseen cases
- Occam's razor – simplicity of representation is preferred



Inducing a decision tree from examples: How? (3)

- How do we find the "simplest" decision tree?
- Intuition: Finding the most "important" attribute and put it on top...



The decision tree algorithm (1)

- If no examples left, return a default value
- If all remaining examples are positive, or all are negative, then can answer YES or NO
- If there are some positive and negative examples but no attributes left to split on, there is noise in the data (one solution is majority vote)
- If there are some positive and negative examples, choose the best attribute to split

The decision tree algorithm (2)

```
function DECISION-TREE-LEARNING(examples, attributes, default) returns a decision tree
  inputs: examples, set of examples
          attributes, set of attributes
          default, default value for the goal predicate

  if examples is empty then return default
  else if all examples have the same classification then return the classification
  else if attributes is empty then return MAJORITY-VALUE(examples)
  else
     best ← CHOOSE-ATTRIBUTE(attributes, examples)
     tree ← a new decision tree with root test best
     for each value v_i of best do
        examples_i ← {elements of examples with best = v_i}
        subtree ← DECISION-TREE-LEARNING(examples_i, attributes − best,
                                          MAJORITY-VALUE(examples))
        add a branch to tree with label v_i and subtree subtree
     end
     return tree
```

ID3: Iterative Dichotomiser 3

- Algorithm used to generate a decision tree
- Invented by Ross Quinlan
- Based on Occam's razor

1. Take all unused attributes and count their entropy concerning test samples
2. Choose attribute for which entropy is minimum
3. Make node containing that attribute

(from Wikipedia, www.wikipedia.org)

Information theory

- In decision trees, the average number of bits of information per output received is *I(P(v_1),...,P(v_m))* when the output symbols v_i, *i*=1,..,*m* occur with probability *P*(v_i).

$$I(P(v_1),\cdots,P(v_m))=\sum_{i=1}^{m}-P(v_i)\log_2 P(v_i)$$

- We usually won't know *P*(v_i), so we estimate it from the training data – just use the relative frequency of occurrence.
- The same information can be provided via ordered combinations of attributes.
- The first attribute is chosen to provide the maximum information, and at lower levels in the tree, the remaining information is provided until all examples at each leaf have the same value for the output (no extra information is needed to get those right).

Information theory (review) (1)

Information content of answer in bits, given probabilities of all possible answers v_i

$$I(P(v_1),\cdots,P(v_m))=\sum_{i=1}^{m}-P(v_i)\log_2 P(v_i)$$

Estimate of information content of a positive answer in a binary classification:

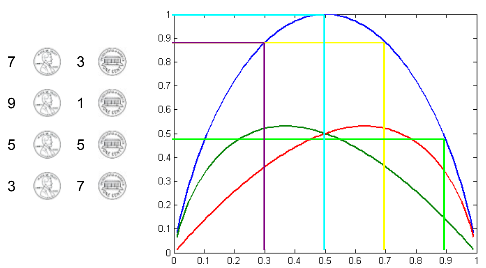
$$I\left(\frac{p}{p+n},\frac{n}{p+n}\right)=-\frac{p}{p+n}\log_2\frac{p}{p+n}-\frac{n}{p+n}\log_2\frac{n}{p+n}$$

where *p* is the number of positive examples and *n* is the number of negative examples.

$$I\left(\frac{1}{2},\frac{1}{2}\right)=-0.5\log_2 0.5-0.5\log_2 0.5=1\text{ bit}$$

Information theory (review) (2)

$$I\left(\frac{p}{p+n},\frac{n}{p+n}\right)=-\frac{p}{p+n}\log_2\frac{p}{p+n}-\frac{n}{p+n}\log_2\frac{n}{p+n}$$



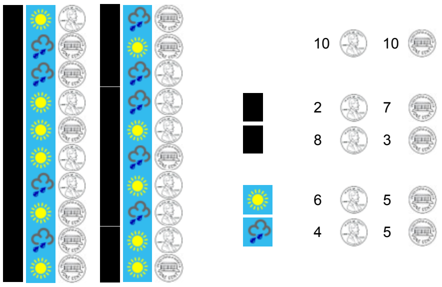
Information *gain* for choosing an attribute (review) (1)

$$Gain(A)=I\left(\frac{p}{p+n},\frac{n}{p+n}\right)-Remainder(A)$$

$$Remainder(A)=\sum_{i=1}^{v}\frac{p_i+n_i}{p+n}I\left(\frac{p_i}{p_i+n_i},\frac{n_i}{p_i+n_i}\right)$$

- Information gain is the difference between the original information requirement and the new requirement
- A comparison of the gain for different attributes gives us the most "important" attribute

Information *gain* for choosing an attribute (review) (2)



Information *gain* for choosing an attribute (review) (3)

$$Remainder(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i})$$

$$Remainder(Gender) = \frac{2+7}{10+10} I(\frac{2}{2+7}, \frac{7}{2+7}) + \frac{8+3}{10+10} I(\frac{8}{8+3}, \frac{3}{8+3})$$

$$= 0.8089$$

$$Remainder(Weather) = \frac{6+5}{10+10} I(\frac{6}{6+5}, \frac{5}{6+5}) + \frac{4+5}{10+10} I(\frac{4}{4+5}, \frac{5}{4+5})$$

$$= 0.9927$$

Information *gain* for choosing an attribute (review) (4)

$$Gain(A) = I(\frac{p}{p+n}, \frac{n}{p+n}) - Remainder(A)$$

$$Gain(Gender) = I(\frac{10}{10+10}, \frac{10}{10+10}) - Remainder(Gender)$$

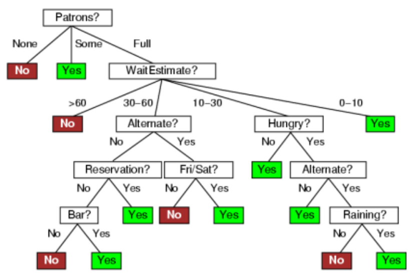
$$= 0.1911$$

$$Gain(Weather) = I(\frac{10}{10+10}, \frac{10}{10+10}) - Remainder(Weather)$$

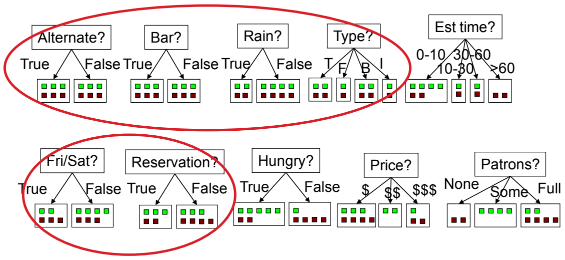
$$= 0.0073$$

The gain from choosing gender to predict the result is greater than the gain from choosing weather to predict the result

A simpler tree?



Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE
FALSE	TRUE	TRUE	FALSE	Full	$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$$$	FALSE	TRUE	Italian	10-30	FALSE
FALSE	FALSE	FALSE	FALSE	None	$	FALSE	FALSE	Thai	0-10	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$	FALSE	FALSE	Burger	30-60	TRUE



An example

Choosing Fri/Sat as the top attribute

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE
FALSE	TRUE	TRUE	FALSE	Full	$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$$$	FALSE	TRUE	Italian	10-30	FALSE
FALSE	FALSE	FALSE	FALSE	None	$	FALSE	FALSE	Thai	0-10	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$	FALSE	FALSE	Burger	30-60	TRUE

Overall: p = n = 6

Fri/Sat="TRUE"
$p_1=2$
$n_1=3$
Fri/Sat="FALSE"
$p_2=4$
$n_2=3$

$$Gain(Fri/Sat) = I(\frac{1}{2}, \frac{1}{2}) - Remainder(Fri/Sat)$$

$$Remainder(Fri/Sat) = \frac{2+3}{6+6} I(\frac{2}{2+3}, \frac{3}{2+3}) + \frac{4+3}{6+6} I(\frac{4}{4+3}, \frac{3}{4+3})$$

$$= \frac{5}{12}(0.9710) + \frac{7}{12}(0.9852)$$

$$Gain(Fri/Sat) = 1 - 0.9793 = 0.0207$$

An example

Choosing Hungry as the top attribute

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE
FALSE	TRUE	TRUE	FALSE	Full	$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$$$	FALSE	TRUE	Italian	10-30	FALSE
FALSE	FALSE	FALSE	FALSE	None	$	FALSE	FALSE	Thai	0-10	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$	FALSE	FALSE	Burger	30-60	TRUE

Overall: p = n = 6

Hungry="TRUE"
p_1=5
n_1=2
Hungry="FALSE"
p_2=1
n_2=4

$$Gain(Hungry) = I(\frac{1}{2},\frac{1}{2}) - Remainder(Hungry)$$

$$Remainder(Hungry) = \frac{5+2}{6+6} I(\frac{5}{5+2},\frac{2}{5+2}) + \frac{1+4}{6+6} I(\frac{1}{1+4},\frac{4}{1+4})$$

$$= \frac{7}{12}(0.8631) + \frac{5}{12}(0.7219)$$

$$Gain(Hungry) = 1 - 0.8043 = 0.1957$$

An example

Using Patrons as the top attribute

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE
FALSE	TRUE	TRUE	FALSE	Full	$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$$$	FALSE	TRUE	Italian	10-30	FALSE
FALSE	FALSE	FALSE	FALSE	None	$	FALSE	FALSE	Thai	0-10	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$	FALSE	FALSE	Burger	30-60	TRUE

$$Gain(Patrons) = I(\frac{1}{2},\frac{1}{2}) - Remainder(Patrons) = 1 - \sum_{i=1}^{3} \frac{p_i+n_i}{p+n} I(\frac{p_i}{p_i+n_i},\frac{n_i}{p_i+n_i})$$

Overall: p = n = 6

Patrons="None"
p_1=0
n_1=2
Patrons="Some"
p_2=4
n_2=0
Patrons="Full"
p_3=2
n_3=4

$$= 1 - \frac{2}{12} I(0,1) - \frac{4}{12} I(1,0) - \frac{6}{12} I(\frac{2}{6},\frac{4}{6})$$

$$= 1 - 0 - 0 - \frac{6}{12}(\frac{2}{6}\log_2\frac{2}{6} - \frac{4}{6}\log_2\frac{4}{6})$$

$$= 1 - \frac{1}{2}(0.9183)$$

$$= 0.5409$$

An example

Using Price as the top attribute

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE
FALSE	TRUE	TRUE	FALSE	Full	$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$$$	FALSE	TRUE	Italian	10-30	FALSE
FALSE	FALSE	FALSE	FALSE	None	$	FALSE	FALSE	Thai	0-10	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$	FALSE	FALSE	Burger	30-60	TRUE

$$Gain(Price) = I(\frac{1}{2},\frac{1}{2}) - Remainder(Price) = 1 - \sum_{i=1}^{3} \frac{p_i+n_i}{p+n} I(\frac{p_i}{p_i+n_i},\frac{n_i}{p_i+n_i})$$

Overall: p = n = 6

Price="$"
p_1=3
n_1=4
Price="$$"
p_2=2
n_2=0
Price="$$$"
p_3=1
n_3=2

$$= 1 - \frac{7}{12} I(\frac{3}{7},\frac{4}{7}) - \frac{2}{12} I(1,0) - \frac{3}{12} I(\frac{1}{3},\frac{2}{3})$$

$$= 1 - \frac{7}{12}(-\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7}) - 0 - \frac{3}{12}(-\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3})$$

$$= 1 - \frac{7}{12}(0.9852) - \frac{3}{12}(0.9183)$$

$$= 0.1597$$

An example

Using Est time as the top attribute

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE
FALSE	TRUE	TRUE	FALSE	Full	$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$$$	FALSE	TRUE	Italian	10-30	FALSE
FALSE	FALSE	FALSE	FALSE	None	$	FALSE	FALSE	Thai	0-10	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$	FALSE	FALSE	Burger	30-60	TRUE

$$Gain(Time) = I(\frac{1}{2},\frac{1}{2}) - Remainder(Time) = 1 - \sum_{i=1}^{4} \frac{p_i+n_i}{p+n} I(\frac{p_i}{p_i+n_i},\frac{n_i}{p_i+n_i})$$

Overall: p = n = 6
Time="0-10"
p_1=4
n_1=2
Time="10-30"
p_2=1
n_2=1
Time="30-60"
p_3=1
n_3=1
Time=">60"
p_4=0
n_4=2

$$= 1 - \frac{6}{12} I(\frac{4}{6},\frac{2}{6}) - \frac{2}{12} I(1,1) - \frac{2}{12} I(1,1) - \frac{2}{12} I(0,1)$$

$$= 1 - \frac{6}{12}(-\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6}) - \frac{2}{12}(1) - \frac{2}{12}(1) - \frac{2}{12}(0)$$

$$= 1 - \frac{6}{12}(0.9183) - \frac{2}{12} - \frac{2}{12}$$

$$= 0.2075$$

An example

Choosing the top attribute

$$Gain(Fri/Sat) = 0.0207$$

$$Gain(Hungry) = 0.1957$$

$$Gain(Patrons) = 0.5409$$

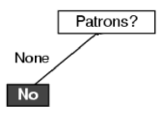
$$Gain(Price) = 0.1597$$

$$Gain(Time) = 0.2075$$

An example

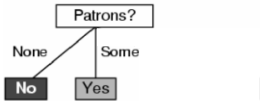
Choosing Patrons as the top attribute (1)

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE
FALSE	TRUE	TRUE	FALSE	Full	$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$$$	FALSE	TRUE	Italian	10-30	FALSE
FALSE	FALSE	FALSE	FALSE	None	$	FALSE	FALSE	Thai	0-10	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$	FALSE	FALSE	Burger	30-60	TRUE



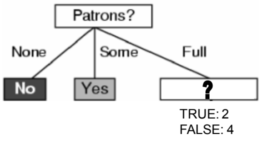
An example
Choosing Patrons as the top attribute (2)

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE
FALSE	TRUE	TRUE	FALSE	Full	$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$$$	FALSE	TRUE	Italian	10-30	FALSE
FALSE	FALSE	FALSE	FALSE	None	$	FALSE	FALSE	Thai	0-10	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$	FALSE	FALSE	Burger	30-60	TRUE



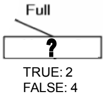
An example
Choosing Patrons as the top attribute (3)

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE
FALSE	TRUE	TRUE	FALSE	Full	$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$$$	FALSE	TRUE	Italian	10-30	FALSE
FALSE	FALSE	FALSE	FALSE	None	$	FALSE	FALSE	Thai	0-10	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$	FALSE	FALSE	Burger	30-60	TRUE



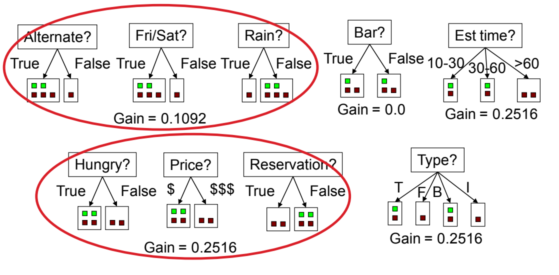
An example
Choosing Patrons as the top attribute (4)

Alternate	Bar	Fri/Sat	Hungry		Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE		$	FALSE	FALSE	Thai	30-60	FALSE
TRUE	FALSE	TRUE	TRUE		$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE		$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	TRUE	FALSE		$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE		$$$	FALSE	TRUE	Italian	10-30	FALSE
TRUE	TRUE	TRUE	TRUE		$	FALSE	FALSE	Burger	30-60	TRUE

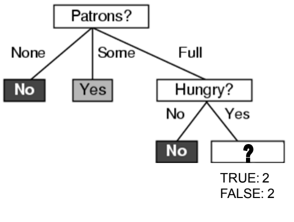


An example
Completing the Decision Tree (1)

Alternate	Bar	Fri/Sat	Hungry	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	$	FALSE	FALSE	Thai	30-60	FALSE
TRUE	FALSE	TRUE	TRUE	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	TRUE	FALSE	$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE	$$$	FALSE	TRUE	Italian	10-30	FALSE
TRUE	TRUE	TRUE	TRUE	$	FALSE	FALSE	Burger	30-60	TRUE

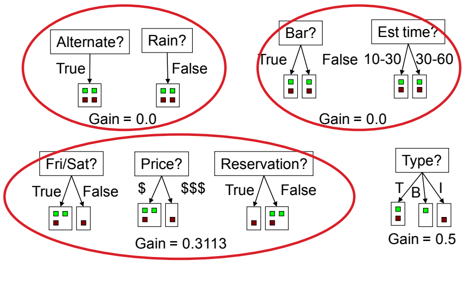


An example
Completing the Decision Tree (2)

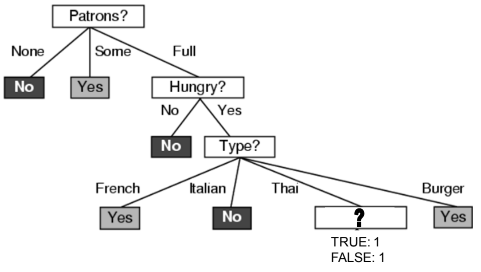


An example
Completing the Decision Tree (3)

Alternate	Bar	Fri/Sat	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	$	FALSE	FALSE	Thai	30-60	FALSE
TRUE	FALSE	TRUE	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	TRUE	TRUE	$$$	FALSE	TRUE	Italian	10-30	FALSE
TRUE	TRUE	TRUE	$	FALSE	FALSE	Burger	30-60	TRUE

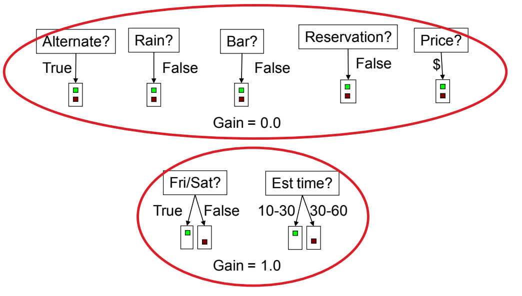


An example
Completing the Decision Tree (4)

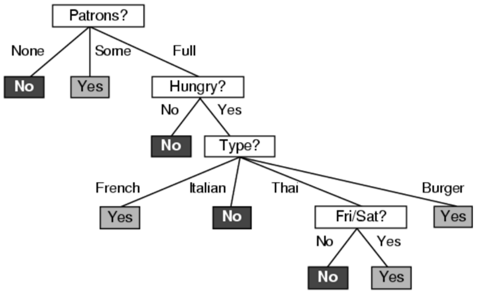


An example
Completing the Decision Tree (5)

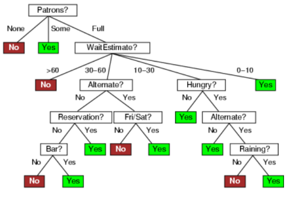
Alternate	Bar	Fri/Sat	Price	Rain	Reservation	Est time	Will wait?
TRUE	FALSE	FALSE	$	FALSE	FALSE	30-60	FALSE
TRUE	FALSE	TRUE	$	FALSE	FALSE	10-30	TRUE

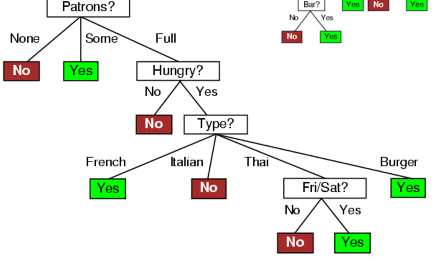


An example
Completing the Decision Tree (6)



A simpler tree





Decision trees: Problems (learning)

- Inconsistent classification of examples (noise), which outcome should be chosen?
  - → Majority vote (parent node)
- May not have seen enough examples for true decision rule to be learned

Decision trees: Problems (representation)

- Decision trees are essentially propositional (one variable, unary predicates)
  - E.g. not possible to represent “choose r_2 if r_2 is cheaper than r_1”
- Good for some functions, really bad for others
  - E.g. parity → large trees that do not generalise
- Continuous input values need to be made discrete (e.g. dividing values into intervals)

Decision Trees: Generalisation

Performance on unseen cases

- Noise
  - Large trees, inconsistent leaves
- Over-fitting, over-specialisation
  - Irrelevant attributes may be used for classification -> avoid using these
- Regularisation and Pruning
  - Can prune based on lack of information gain

Decision Tree Pruning

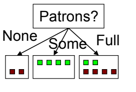
- Probability that, given there is no underlying pattern, a sample of a given size would exhibit the observed deviation from the expected distribution of positive and negative examples
- χ^2 pruning
- Comparing the actual numbers of positive and negative examples in a subset to the expected numbers of positive and negative examples

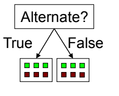
Decision Tree Pruning: Variables

- For a given attribute with *v* possible values and *p*+*n* examples remaining to be classified
- *p* = number of positive examples
- *n* = number of negative examples
- p_i = number of positive examples in the subset for attribute value *i*
- n_i = number of negative examples in the subset for attribute value *i*
- $\hat{p}_i$ = expected number of positive examples in the subset for attribute value *i*
- $\hat{n}_i$ = expected number of negative examples in the subset for attribute value *i*
- *D* = total deviation

Decision Tree Pruning: Restaurant Example

- $p = 6$, $n = 6$
- Patrons
  - $p_{none} = 0$, $n_{none} = 2$
  - $p_{some} = 4$, $n_{some} = 0$
  - $p_{full} = 2$, $n_{full} = 4$
- Alternate
  - $p_{true} = 3$, $n_{true} = 3$
  - $p_{false} = 3$, $n_{false} = 3$





Decision Tree Pruning: Equations

$$\hat{p}_i = p \times \frac{p_i + n_i}{p + n} \qquad \hat{n}_i = n \times \frac{p_i + n_i}{p + n}$$

$$D = \sum_{i=1}^{v} \frac{(p_i - \hat{p}_i)^2}{\hat{p}_i} + \frac{(n_i - \hat{n}_i)^2}{\hat{n}_i}$$

Decision Tree Pruning: χ^2 pruning

- The probability that an attribute is irrelevant can be calculated by using χ^2 tables or statistical software
- Means that noise in the training data can be tolerated
- Pruned trees can perform better when the data is very noisy
- Pruned trees are often smaller and easier to understand

Current best learning

- Given a set of examples, find the best hypothesis / set of hypotheses that matches, with an appropriate degree of generalisation vs specialisation
- Maintain a single hypothesis, adjust as new examples are available by
  - generalising or
  - specialising

Learning with prior knowledge (1)

- Examples: described by attributes
- Hypothesis: predicts that a certain set of examples (that satisfy its definition) will be examples of the goal predicate
- Hypothesis space **H** is the set of all hypotheses $\{H_1,\ldots,H_n\}$
- As examples are considered, hypotheses that are not consistent are ruled out

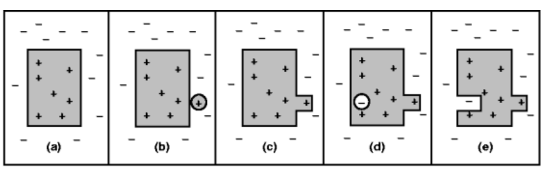
Learning with prior knowledge (2)

- Learning = finding a hypothesis that agrees with the observed examples
- A hypothesis that does not agree with observed examples may be:
  - False positives (hypothesis says the example should be positive, but it is negative)
  - False negatives (hypothesis says the example should be negative, but it is positive)

Searching for a logic-based hypothesis

- Assume representation is predicate-based
- A hypothesis defines a goal predicate
  - which is true for all “positives” and false for all “negatives”
- The hypothesis predicts a set of examples as “positives” – the extension of the hypothesis

Searching the hypothesis space



Training examples: H_r is illustrated as the boundary between its positives (extension) and its negatives. (a) consistent, (b) one false negative, (c) H_r is generalized, (d) one false positive, (e) H_r is specialized.

Current best learning

- Maintain a single hypothesis that is adjusted as new examples are encountered
- False negative -> generalise
  - (increase the extension of the hypothesis to include this example)
- False positive -> specialise
  - (decrease the extension of the hypothesis to exclude this example)
- In each update, all previous examples must be checked for consistency

Current best learning

Sample	Sky {Sunny, Cloudy, Rainy}	Temperature {Warm, Cold}	Humidity {Low, Normal, High}	Wind {Weak, Strong}	Happy {Yes, No}
1	Sunny	Warm	Normal	Strong	Yes
2	Rainy	Warm	High	Strong	No
3	Cloudy	Cold	Normal	Weak	Yes
4	Sunny	Cold	Normal	Weak	Yes

H1= < Cloudy, Warm, ?, ? > False negative, generalise
H2= < ?, Warm, ?, ? > False positive, specialise
H3= < ?, Warm, Normal, ? > False negative, generalise
H4= < ?, ?, Normal, ? > OK!

Current best learning: Algorithm

function CURRENT-BEST-LEARNING(*examples*) **returns** a hypothesis

H ← any hypothesis consistent with the first example in *examples*
for each remaining example in *examples* **do**
  if *e* is false positive for *H* **then**
    H ← **choose** a specialization of *H* consistent with *examples*
  else if *e* is false negative for *H* **then**
    H ← **choose** a generalization of *H* consistent with *examples*
  if no consistent specialization/generalization can be found **then fail**
end
return *H*

Inducing a classifier from examples

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE
FALSE	TRUE	TRUE	FALSE	Full	$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$$$	FALSE	TRUE	Italian	10-30	FALSE
FALSE	FALSE	FALSE	FALSE	None	$	FALSE	FALSE	Thai	0-10	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$	FALSE	FALSE	Burger	30-60	TRUE

- Attributes (features)
- Example
- Classification

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE
FALSE	TRUE	TRUE	FALSE	Full	$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$$$	FALSE	TRUE	Italian	10-30	FALSE
FALSE	FALSE	FALSE	FALSE	None	$	FALSE	FALSE	Thai	0-10	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$	FALSE	FALSE	Burger	30-60	TRUE

h_1: $\forall x$ WillWait(x) $\Leftrightarrow$ Alternate(x)

Alternate(X1) $\wedge$ $\neg$Bar(X1) $\wedge$ $\neg$Fri/Sat(X1) $\wedge$ Hungry(X1)... WillWait(X1)

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE
FALSE	TRUE	TRUE	FALSE	Full	$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$$$	FALSE	TRUE	Italian	10-30	FALSE
FALSE	FALSE	FALSE	FALSE	None	$	FALSE	FALSE	Thai	0-10	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$	FALSE	FALSE	Burger	30-60	TRUE

h_1: $\forall x$ WillWait(x) $\Leftrightarrow$ Alternate(x)

h_2: $\forall x$ WillWait(x) $\Leftrightarrow$ Alternate(x) $\wedge$ Patrons(x, Some)

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE
FALSE	TRUE	TRUE	FALSE	Full	$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$$$	FALSE	TRUE	Italian	10-30	FALSE
FALSE	FALSE	FALSE	FALSE	None	$	FALSE	FALSE	Thai	0-10	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$	FALSE	FALSE	Burger	30-60	TRUE

h_1: $\forall x$ WillWait(x) $\Leftrightarrow$ Alternate(x)

h_2: $\forall x$ WillWait(x) $\Leftrightarrow$ Alternate(x) $\wedge$ Patrons(x, Some)

h_3: $\forall x$ WillWait(x) $\Leftrightarrow$ Patrons(x, Some)

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE
FALSE	TRUE	TRUE	FALSE	Full	$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$$$	FALSE	TRUE	Italian	10-30	FALSE
FALSE	FALSE	FALSE	FALSE	None	$	FALSE	FALSE	Thai	0-10	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$	FALSE	FALSE	Burger	30-60	TRUE

h_1: $\forall x \; \mathrm{WillWait}(x) \Leftrightarrow \mathrm{Alternate}(x)$

h_2: $\forall x \; \mathrm{WillWait}(x) \Leftrightarrow \mathrm{Alternate}(x) \wedge \mathrm{Patrons}(x, \mathrm{Some})$

h_3: $\forall x \; \mathrm{WillWait}(x) \Leftrightarrow \mathrm{Patrons}(x, \mathrm{Some})$

h_4: $\forall x \; \mathrm{WillWait}(x) \Leftrightarrow \mathrm{Patrons}(x, \mathrm{Some}) \vee [\mathrm{Patrons}(x, \mathrm{Full}) \wedge \mathrm{Fri/Sat}(x)]$

…

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE

h_1: <True,*,*,*,*,*,*,*,*,*>

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE

False positive -> specialise

h_1: <True,*,*,*,*,*,*,*,*,*>

h_2: <True,*,*,*,Some,*,*,*,*,*>

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE

False negative -> generalise

h_1: <True,*,*,*,*,*,*,*,*,*>

h_2: <True,*,*,*,Some,*,*,*,*,*>

h_3: <*,*,*,*,Some,*,*,*,*,*>

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE

False negative -> generalise

h_1: <True,*,*,*,*,*,*,*,*,*>

h_2: <True,*,*,*,Some,*,*,*,*,*>

h_3: <*,*,*,*,Some,*,*,*,*,*>

h_4: <*,*,*,*,Some,*,*,*,*,*> OR <*,*,*,*,Full,*,*,*,*,10-30>

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE

OK!

h_1: <True,*,*,*,*,*,*,*,*,*>

h_2: <True,*,*,*,Some,*,*,*,*,*>

h_3: <*,*,*,*,Some,*,*,*,*,*>

h_4: <*,*,*,*,Some,*,*,*,*,*> OR <*,*,*,*,Full,*,*,*,*,10-30>

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE

OK!

h_1: <True,*,*,*,*,*,*,*,*,*>

h_2: <True,*,*,*,Some,*,*,*,*,*>

h_3: <*,*,*,*,Some,*,*,*,*,*>

h_4: <*,*,*,*,Some,*,*,*,*,*> OR <*,*,*,*,Full,*,*,*,*,10-30>

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE

OK!

h_1: <True,*,*,*,*,*,*,*,*,*>

h_2: <True,*,*,*,Some,*,*,*,*,*>

h_3: <*,*,*,*,Some,*,*,*,*,*>

h_4: <*,*,*,*,Some,*,*,*,*,*> OR <*,*,*,*,Full,*,*,*,*,10-30>

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE

OK!

h_1: <True,*,*,*,*,*,*,*,*,*>

h_2: <True,*,*,*,Some,*,*,*,*,*>

h_3: <*,*,*,*,Some,*,*,*,*,*>

h_4: <*,*,*,*,Some,*,*,*,*,*> OR <*,*,*,*,Full,*,*,*,*,10-30>

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE
FALSE	TRUE	TRUE	FALSE	Full	$	TRUE	FALSE	Burger	>60	FALSE

OK!

h_1: <True,*,*,*,*,*,*,*,*,*>

h_2: <True,*,*,*,Some,*,*,*,*,*>

h_3: <*,*,*,*,Some,*,*,*,*,*>

h_4: <*,*,*,*,Some,*,*,*,*,*> OR <*,*,*,*,Full,*,*,*,*,10-30>

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE
FALSE	TRUE	TRUE	FALSE	Full	$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$$$	FALSE	TRUE	Italian	10-30	FALSE

False positive -> specialise

h_1: <True,*,*,*,*,*,*,*,*,*>

h_2: <True,*,*,*,Some,*,*,*,*,*>

h_3: <*,*,*,*,Some,*,*,*,*,*>

h_4: <*,*,*,*,Some,*,*,*,*,*> OR <*,*,*,*,Full,*,*,*,*,10-30>

h_5: <*,*,*,*,Some,*,*,*,*,*> OR <*,*,*,*,Full,*,*,False,*,10-30>

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE
FALSE	TRUE	TRUE	FALSE	Full	$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$$$	FALSE	TRUE	Italian	10-30	FALSE
FALSE	FALSE	FALSE	FALSE	None	$	FALSE	FALSE	Thai	0-10	FALSE

OK!

h_1: <True,*,*,*,*,*,*,*,*,*>

h_2: <True,*,*,*,Some,*,*,*,*,*>

h_3: <*,*,*,*,Some,*,*,*,*,*>

h_4: <*,*,*,*,Some,*,*,*,*,*> OR <*,*,*,*,Full,*,*,*,*,10-30>

h_5: <*,*,*,*,Some,*,*,*,*,*> OR <*,*,*,*,Full,*,*,False,*,10-30>

Alternate	Bar	Fri/Sat	Hungry	Patrons	Price	Rain	Reservation	Type	Est time	Will wait?
TRUE	FALSE	FALSE	TRUE	Some	$$$	FALSE	TRUE	French	0-10	TRUE
TRUE	FALSE	FALSE	TRUE	Full	$	FALSE	FALSE	Thai	30-60	FALSE
FALSE	TRUE	FALSE	FALSE	Some	$	FALSE	FALSE	Burger	0-10	TRUE
TRUE	FALSE	TRUE	TRUE	Full	$	FALSE	FALSE	Thai	10-30	TRUE
TRUE	FALSE	TRUE	FALSE	Full	$$$	FALSE	TRUE	French	>60	FALSE
FALSE	TRUE	FALSE	TRUE	Some	$$	TRUE	TRUE	Italian	0-10	TRUE
FALSE	TRUE	FALSE	FALSE	None	$	TRUE	FALSE	Burger	0-10	FALSE
FALSE	FALSE	FALSE	TRUE	Some	$$	TRUE	TRUE	Thai	0-10	TRUE
FALSE	TRUE	TRUE	FALSE	Full	$	TRUE	FALSE	Burger	>60	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$$$	FALSE	TRUE	Italian	10-30	FALSE
FALSE	FALSE	FALSE	FALSE	None	$	FALSE	FALSE	Thai	0-10	FALSE
TRUE	TRUE	TRUE	TRUE	Full	$	FALSE	FALSE	Burger	30-60	TRUE

False negative -> generalise

h_1: <True,*,*,*,*,*,*,*,*,*>

h_2: <True,*,*,*,Some,*,*,*,*,*>

h_3: <*,*,*,*,Some,*,*,*,*,*>

h_4: <*,*,*,*,Some,*,*,*,*,*> OR <*,*,*,*,Full,*,*,*,*,10-30>

h_5: <*,*,*,*,Some,*,*,*,*,*> OR <*,*,*,*,Full,*,*,False,*,10-30>

h_6: <*,*,*,*,Some,*,*,*,*,*> OR <*,*,*,*,Full,*,*,False,*,10-30>
OR <*,*,*,*,Full,*,*,*,Burger,30-60>

Current best learning: Problems (learning) (1)

- Note that at any point there may be several possible specialisations or generalisations, but only one can be chosen
- This may lead to a situation where no simple specialisation or generalisation will fit the examples
- In this case, backtrack to a previous choice
- Also, the choices may not lead to the simplest hypothesis

Current best learning: Problems (learning) (2)

- Checking previous examples for each update is expensive
- Search process may involve a lot of backtracking, as has to choose a particular hypothesis
- Noise in the examples may mean that a consistent hypothesis cannot be found

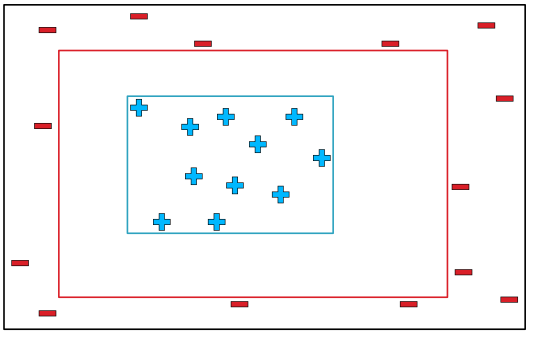
Current best learning: Problems (representation)

- As for decision trees
- Propositional (one variable, unary predicates)
  - E.g. not possible to represent “choose r_2 if r_2 is cheaper than r_1”
- Good for some functions, really bad for others
  - E.g. parity → large trees that do not generalise
- Continuous input values need to be made discrete (e.g. dividing values into intervals)

Alternative to current best learning

- Version space learning
  - Starts with the entire hypothesis space, inconsistent hypotheses are ruled out
  - Starts from both most general and most specific
  - In between these boundaries are the consistent hypotheses

Version Space Learning



Version Space Learning

- Maintain 2 sets of hypotheses / Boundary Sets that are updated as new examples are encountered
  - General Boundary (G-set)
  - Specific Boundary (S-set)
- False positive for S_i -> S_i too general, but no consistent specialisations, so throw out
- False negative for S_i -> generalise
- False positive for G_i -> specialise
- False negative for G_i -> G_i too specific, but no consistent generalisations, so throw out

Version Space Learning: Problems

- If noise or insufficient attributes for exact classification in the examples, version space will collapse
- For some hypothesis spaces, the number of elements in the boundary sets may grow exponentially in the number of attributes

Summary

- Decision tree learning
- Current best learning
- Version space learning