Logic

FOL Syntax

$$
\begin{aligned}
\textit{Sentence} &\rightarrow \textit{AtomicSentence} \mid \textit{ComplexSentence} \\
\textit{AtomicSentence} &\rightarrow \textit{Predicate} \mid \textit{Predicate}(\textit{Term}, \ldots) \mid \textit{Term} = \textit{Term} \\
\textit{ComplexSentence} &\rightarrow (\ \textit{Sentence}\ ) \mid [\ \textit{Sentence}\ ] \\
&\mid \neg\ \textit{Sentence} \\
&\mid \textit{Sentence} \wedge \textit{Sentence} \\
&\mid \textit{Sentence} \vee \textit{Sentence} \\
&\mid \textit{Sentence} \Rightarrow \textit{Sentence} \\
&\mid \textit{Sentence} \Leftrightarrow \textit{Sentence} \\
&\mid \textit{Quantifier}\ \textit{Variable}, \ldots\ \textit{Sentence} \\
\textit{Term} &\rightarrow \textit{Function}(\textit{Term}, \ldots) \\
&\mid \textit{Constant} \\
&\mid \textit{Variable} \\
\textit{Quantifier} &\rightarrow \forall \mid \exists \\
\textit{Constant} &\rightarrow A \mid X_1 \mid \textit{John} \mid \cdots \\
\textit{Variable} &\rightarrow a \mid x \mid s \mid \cdots \\
\textit{Predicate} &\rightarrow \textit{True} \mid \textit{False} \mid \textit{After} \mid \textit{Loves} \mid \textit{Raining} \mid \cdots \\
\textit{Function} &\rightarrow \textit{Mother} \mid \textit{LeftLeg} \mid \cdots \\
\text{OPERATOR PRECEDENCE} &: \neg, =, \wedge, \vee, \Rightarrow, \Leftrightarrow
\end{aligned}
$$

Rules of Inference

1. Modus Ponens (M.P.)	$1.\ p \Rightarrow q \quad 2.\ p \quad \therefore q$
2. Modus Tollens (M.T.)	$1.\ p \Rightarrow q \quad 2.\ \sim q \quad \therefore \sim p$
3. Hypothetical Syllogism (H.S.)	$1.\ p \Rightarrow q \quad 2.\ q \Rightarrow r \quad \therefore p \Rightarrow r$
4. Disjunctive Syllogism (D.S.)	$1.\ p \vee q \quad 2.\ \sim p \quad \therefore q$
5. Constructive Dilemma (C.D.)	$1.\ (p \Rightarrow q) . (r \Rightarrow s) \quad 2.\ p \vee r \quad \therefore q \vee s$
7. Simplification (Simp.)	$1.\ p \wedge q \quad \therefore p$
8. Conjunction (Conj.)	$1.\ p \quad 2.\ q \quad \therefore p \wedge q$
9. Addition (Add.)	$1.\ p \quad \therefore p \vee q$

FOL Rules (Copi)

Rules of Substitution

10. De Morgan's Theorem (De M.)	$\sim(p \wedge q) \equiv (\sim p \vee \sim q)$	$\sim(p \vee q) \equiv (\sim p \wedge \sim q)$
11. Commutation (Com.)	$(p \vee q) \equiv (q \vee p)$	$(p \wedge q) \equiv (q \wedge p)$
12. Association (Assoc.)	$[p \vee (q \vee r)] \equiv [(p \vee q) \vee r]$	$[p \wedge (q \wedge r)] \equiv [(p \wedge q) \wedge r]$
13. Distribution (Dist)	$[p \wedge (q \vee r)] \equiv [(p \wedge q) \vee (p \wedge r)]$	$[p \vee (q \wedge r)] \equiv [(p \vee q) \wedge (p \vee r)]$
14.Double Negation (D.N.)	$p \equiv \sim\sim p$	
15. Transposition (Trans.)	$(p \Rightarrow q) \equiv (\sim q \Rightarrow \sim p)$	
16. Material Implication (M. Imp.)	$(p \Rightarrow q) \equiv (\sim p \vee q)$	
17. Material Equivalence (M. Equiv.)	$(p \equiv q) \equiv [(p \Rightarrow q) \wedge (q \Rightarrow p)]$	$(p \equiv q) \equiv [(p \wedge q) \vee (\sim p \wedge \sim q)]$
18. Exportation (Exp.)	$[(p \wedge q) \Rightarrow r] \equiv [p \Rightarrow (q \Rightarrow r)]$	
19. Tautology (Taut.)	$p \equiv (p \vee p)$	$p \equiv (p \wedge p)$

1. Courses are either tough or boring.
2. Not all courses are boring.
3. Therefore there are tough courses.

(Cx, Tx, Bx,)

Dealing with Time

Translate into first-order logic :

- He left town in the morning
- Understanding leads to friendship

Russell & Norvig 3d ed:

8.1 to 8.4
9.1, 9.2 (skip 9.2.3 to 9.4),
9.5 resolution (skip 9.5.4 - 9.5.5)

4.5 Show that $(\exists z)(\forall x)[P(x) \Rightarrow Q(z)]$ and $(\exists z)[(\exists x)P(x) \Rightarrow Q(z)]$ are equivalent.

4.6 Convert the following wffs to clause form:

(a) $(\forall x)[P(x) \Rightarrow P(x)]$

(b) $\{\sim\{(\forall x)P(x)\}\} \Rightarrow (\exists x)[\sim P(x)]$

(c) $\sim(\forall x)\{P(x) \Rightarrow \{(\forall y)[P(y) \Rightarrow P(f(x,y))]$
$\wedge \sim(\forall y)[Q(x,y) \Rightarrow P(y)]\}\}$

(d) $(\forall x)(\exists y)$
$\{[P(x,y) \Rightarrow Q(y,x)] \wedge [Q(y,x) \Rightarrow S(x,y)]\}$
$\Rightarrow (\exists x)(\forall y)[P(x,y) \Rightarrow S(x,y)]$

4.7 Show by an example that the composition of substitutions is not commutative.

4.8 Show that resolution is sound; that is, show that the resolvent of two clauses logically follows from the two clauses.

4.9 Find the mgu of the set $\{P(x,z,y), P(w,u,w), P(A,u,u)\}$.

4.10 Explain why the following sets of literals do not unify:

(a) $\{P(f(x,x),A), P(f(y,f(y,A)),A)\}$

(b) $\{\sim P(A), P(x)\}$

(c) $\{P(f(A),x), P(x,A)\}$

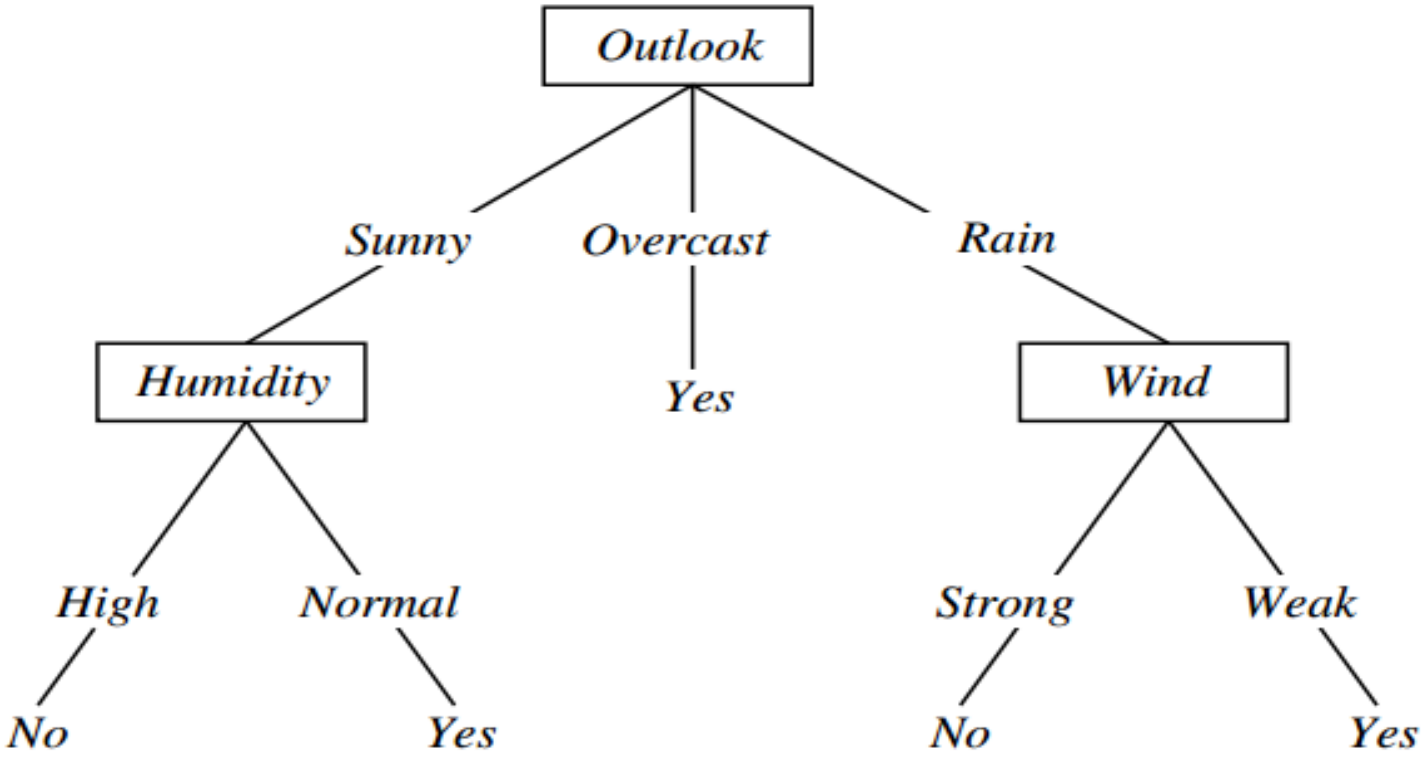
Problems from Nilsson 82

Learning Logical Rules
Decision Trees

Duda and Hart, Ch.1

Russell & Norvig Ch. 18

Boolean Decision Trees



Attribute-based representations

- Examples described by attribute values (Boolean, discrete, continuous)
- E.g., situations where I will/won't wait for a table:

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*Wait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

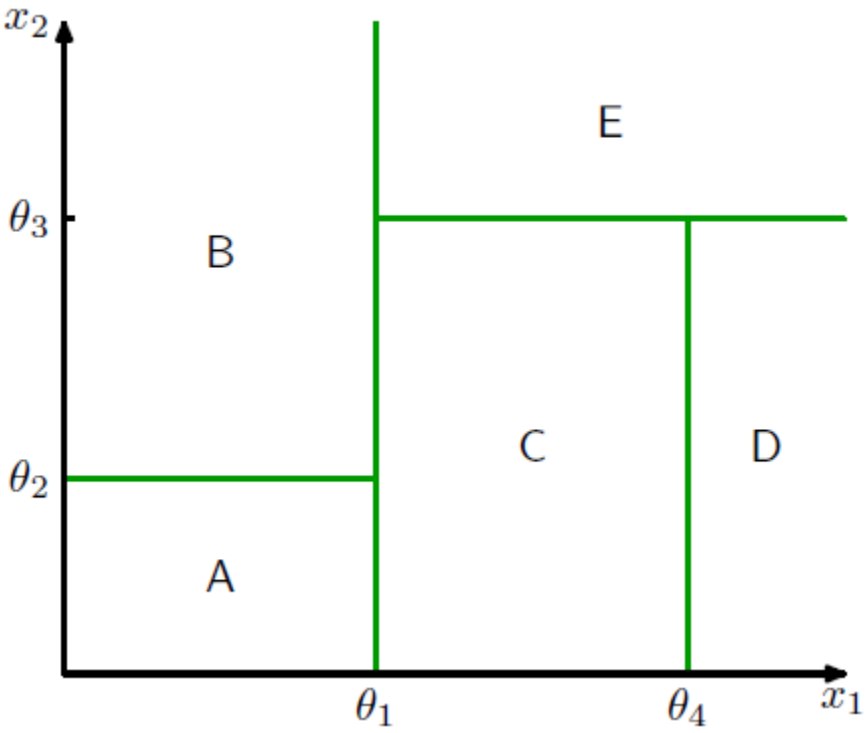
- Classification of examples is positive (T) or negative (F)

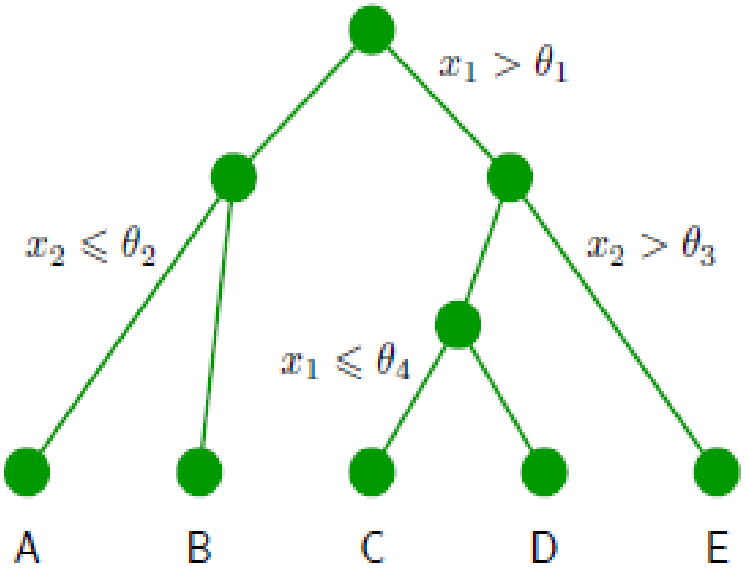
Learning decision trees

Problem: decide whether to wait for a table at a restaurant, based on the following attributes:

1. Alternate: is there an alternative restaurant nearby?
2. Bar: is there a comfortable bar area to wait in?
3. Fri/Sat: is today Friday or Saturday?
4. Hungry: are we hungry?
5. Patrons: number of people in the restaurant (None, Some, Full)
6. Price: price range ($, $$, $$$)
7. Raining: is it raining outside?
8. Reservation: have we made a reservation?
9. Type: kind of restaurant (French, Italian, Thai, Burger)
10. WaitEstimate: estimated waiting time (0-10, 10-30, 30-60, >60)

Continuous orthogonal domains



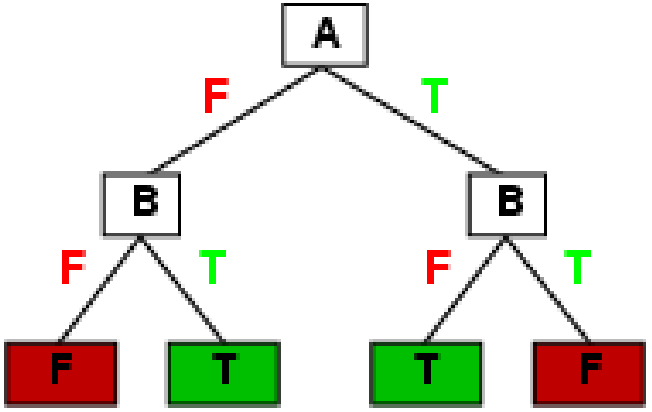


classification and regression trees CART
[Breiman 84] ID3: [Quinlan 86]

Expressiveness

- Decision trees can express any function of the input attributes.
- E.g., for Boolean functions, truth table row → path to leaf:

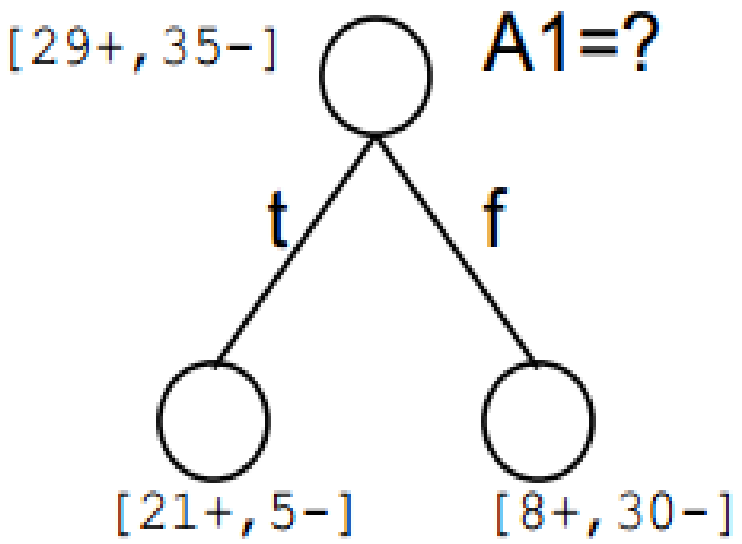
A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F

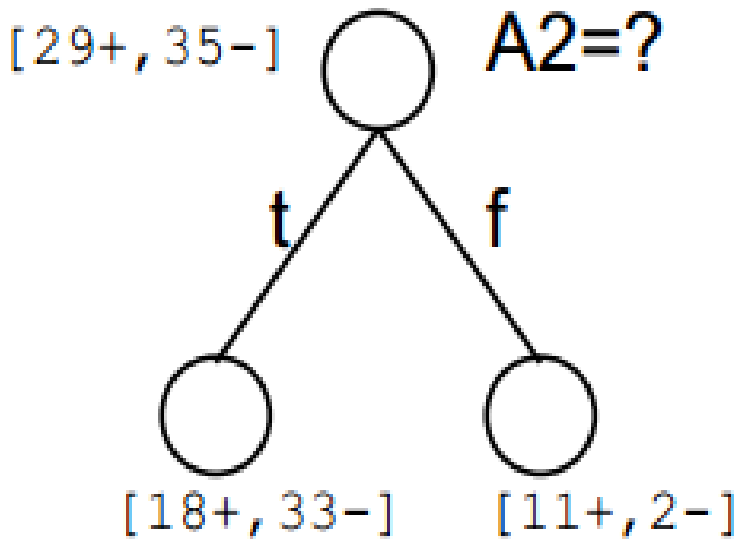


- Trivially, there is a consistent decision tree for any training set with one path to leaf for each example (unless *f* nondeterministic in *x*) but it probably won't generalize to new examples

- Prefer to find more compact decision trees

Which attribute to use first?

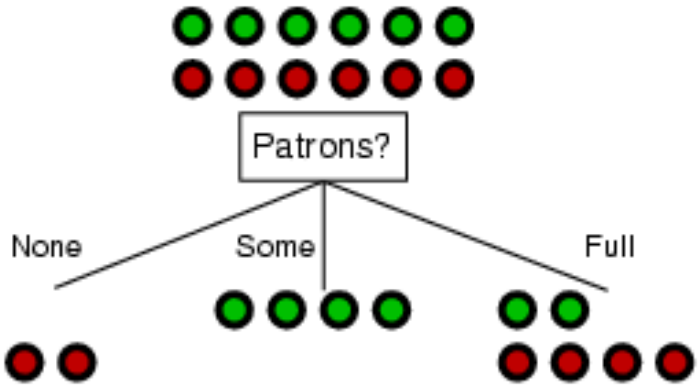


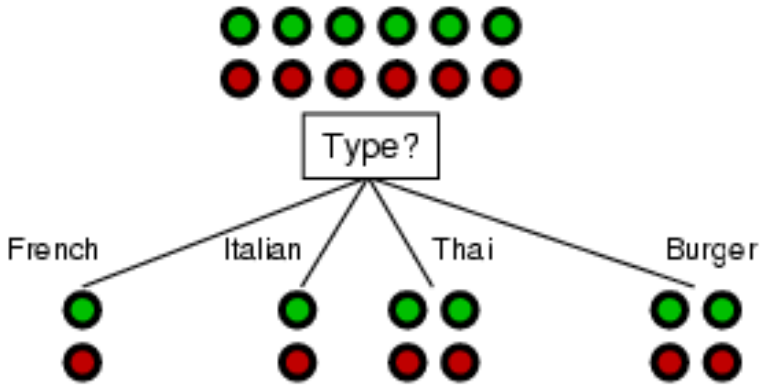


Gain(S; A) = expected reduction in entropy due to sorting on attribute A

Choosing an attribute

- **Idea: a good attribute splits the examples into subsets that are (ideally) "all positive" or "all negative"**





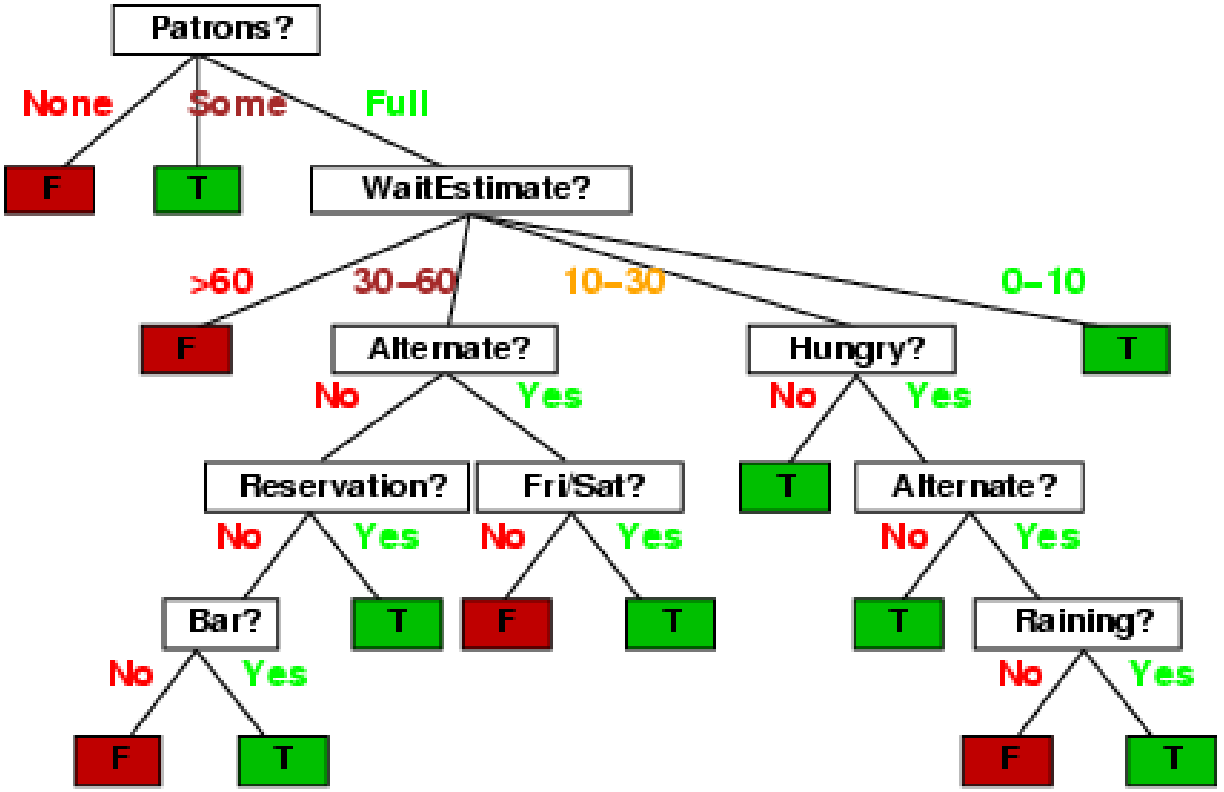
- ***Gain (Patrons)*** $= B(\frac{6}{12}) - [\frac{2}{12} B(\frac{0}{2}) + \frac{4}{12} B(\frac{4}{4}) + \frac{6}{12} B(\frac{2}{2})] = 0.541 bits$

- ***Gain (Type)*** $= 1 - [1.B(\frac{1}{2})] = 0$

Information Gain is higher for Patrons

Decision trees

- One possible representation for hypotheses
- E.g., here is the "true" tree for deciding whether to wait:



Information gain

- **A chosen attribute *A* divides the training set *E* into subsets $E_1, \ldots, E_v$ according to their values for *A*, where *A* has v distinct values.**

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

- **Information Gain (IG) or reduction in entropy from the attribute test:**

$$IG(A) = B\left(\frac{p}{p + n}\right) - remainder(A)$$

- **Choose the attribute with the largest IG**

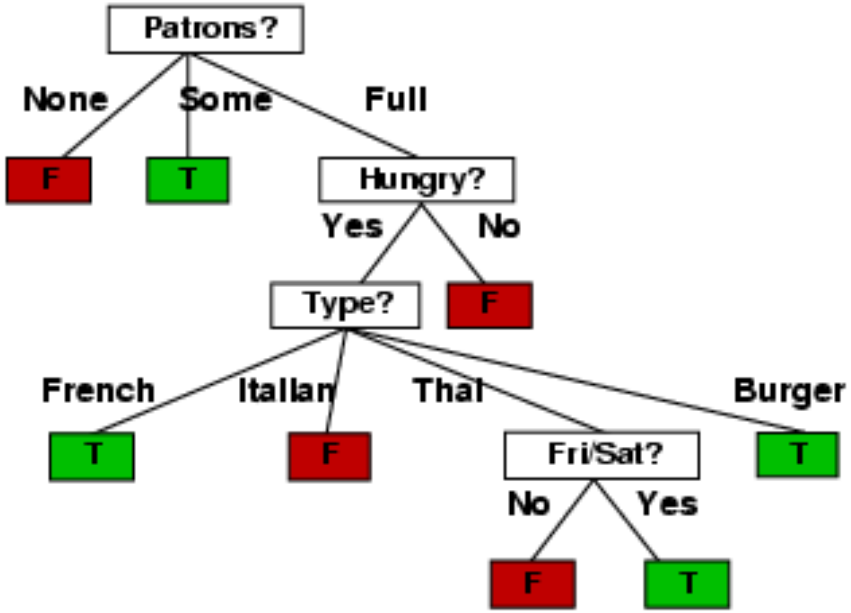
Decision tree learning

- **Aim: find a small tree consistent with the training examples**
- **Idea: (recursively) choose "most significant" attribute as root of (sub)tree**

```
function DTL(examples, attributes, default) returns a decision tree
    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 MODE(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 ← DTL(examples_i, attributes − best, MODE(examples))
            add a branch to tree with label v_i and subtree subtree
        return tree
```

Example contd.

- Decision tree learned from the 12 examples:



- Substantially simpler than “true” tree---a more complex hypothesis isn’t justified by small amount of data

Too many ways to order the tree

How many distinct decision trees with *n* Boolean attributes?

= number of Boolean functions

= number of distinct truth tables with 2^n rows = 2^{2^n}

- **E.g., with 6 Boolean attributes, there are 18,446,744,073,709,551,616 trees**

Hypothesis spaces

How many distinct decision trees with *n* Boolean attributes?

= number of Boolean functions

= number of distinct truth tables with 2^n rows = 2^{2^n}

- **E.g., with 6 Boolean attributes, there are 18,446,744,073,709,551,616 trees**

How many purely conjunctive hypotheses (e.g., *Hungry* $\wedge \neg$*Rain*)?

- **Each attribute can be in (positive), in (negative), or out**
  $\Rightarrow 3^n$ distinct conjunctive hypotheses
- **More expressive hypothesis space**
  - increases chance that target function can be expressed
  - increases number of hypotheses consistent with training set
    $\Rightarrow$ may get worse predictions

Inductive Logic Programming

Review

- **literal** : atomic formula or its negation
- **clause** : disjunction of literals
- **CNF** : conjunction of clauses

Horn Clauses

- **Horn clause** : clause with single positive literal
- E.g. $\sim p \vee \sim q \vee \sim r \vee s$

  or equivalently:

  $$p \wedge q \wedge r \Rightarrow s$$
- useful **restriction** on full first-order logic : propositional Horn clauses are **satisfiable** in polynomial time (HORNSAT)
- General proposition SAT: NP-complete

Prolog

- Prolog = Programming with Logic

- Works with Horn clauses

  ~p V ~q V ~r V s written as

  s :- p, q, r

- read as *s if* p and q and r
- Resolving two Horn clauses always results in a Horn clause

Example

- Set of Horn clauses :

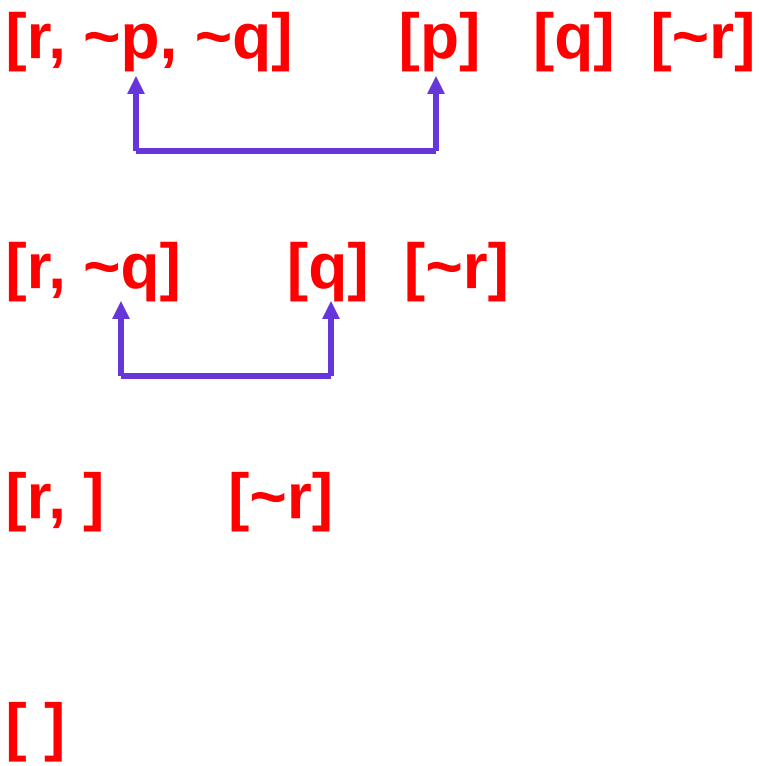
  [r,~p,~q] [p] [q]

- Goal: to prove **r**

  → Add goal negation **~r**

  [r,~p,~ q] [a] [q] [~r]

Resolution refutation



Logic Programming

- **example of a logic program:**

```
parent_of(charles,george).
parent_of(george,diana).
parent_of(bob,harry).
parent_of(harry,elizabeth).

grandparent_of(X,Y) :- parent_of(X,Z),
                       parent_of(Z,Y).
```

- From the program, we can ask queries about grandparents.
- **Query:** `grandparent_of(X,Y)?`
- Answers:
  - `grandparent_of(charles,diana).`
  - `grandparent_of(bob,elizabeth).`

Forms of Reasoning

- **Deduction:** **From causes to effect** **(Prediction)**
  - fact a, rule a => b

    INFER b (*First-order logic*)
- **Abduction:** **From effects to possible causes** **(Explanation)**
  - rule a => b, observe b

    AN EXPLANATION a
- **Induction:** **From correlated observations to rules** **(Learning)**
  - observe correlation between a_1, b_1, ... a_n, b_n

    LEARN RULE a –> b

Inductive Logic Programming : Progol

Given:

Background knowledge (of the domain): facts

Examples (of the relation to be learned): facts

Try to learn

Theories (as a result of learning): rules

ILP : a *form of machine learning* where both the data and the hypotheses are logical expressions

ILP – formal definitions

- **Given**
  - a logic program B representing background knowledge
  - a set of positive examples E^+
  - a set of negative examples E^-
- **Find hypothesis H such that:**

1. $B \cup H \models e$ for every $e \in E^+$.
2. $B \cup H \not\models f$ for every $f \in E^-$.
3. $B \cup H$ is consistent.

Assume that $B \not\models e$ for some $e \in E^+$.

Example

- **Background knowledge *B*:**
  - `parent_of(charles,george).`
  - `parent_of(george,diana).`
  - `parent_of(bob,harry).`
  - `parent_of(harry,elizabeth).`
- **Positive examples E^+:**
  - `grandparent_of(charles,diana).`
  - `grandparent_of(bob,elizabeth).`
- **Generate hypothesis *H*:**
  - `grandparent_of(X,Y) :- parent_of(X,Z), parent_of(Z,Y).`

Rule Learning (Intuition)

- **How to come up with a rule for `grandparent_of(X,Y)`?**
  1. Take the example `grandparent_of(bob,elizabeth).`
  2. Find the subset of background knowledge relevant to this example: `parent_of(bob,harry),`
     `parent_of(harry,elizabeth).`
  3. Form a rule from these facts

     ```
     grandparent_of(bob,elizabeth) :-
      parent_of(bob,harry), parent_of(harry,elizabeth).
     ```
  4. Generalize the rule

     ```
     grandparent_of(X,Y) :- parent_of(X,Z), parent_of(Z,Y).
     ```
  5. Check if this rule is valid w.r.t the positive and the negative examples

Progol Algorithm Outline

1. From a subset of positive examples, construct the **most specific rule r_s.**
2. Based on r_s, find a **generalized form r_g** of r_s so that *score*(r_g) has the highest value among all candidates.
3. Remove all positive examples that are covered by r_g.
4. Go to step 1 if there are still positive examples that are not yet covered.

Scoring hypotheses

- ***score*(*r*) is a measure of how well a rule *r* explains all the examples with preference given to shorter rules.**
  - p_r = number of +ve examples correctly deducible from *r*
  - n_r = number of -ve examples correctly deducible from *r*
  - c_r = number of body literals in rule *r*
  - $score(r) = p_r - (n_r + c_r)$

Decision Theory

Decision Theory

Inference step

Determine either $p(t|\mathbf{x})$ or $p(\mathbf{x}, t)$.

Decision step

For given x, determine optimal t.

Minimum Expected Loss

Example: classify medical images as ‘cancer’ or ‘normal’

Loss matrix L:

$$
\begin{array}{cc} & \text{Decision} \\ & \begin{array}{cc} \text{cancer} & \text{normal} \end{array} \\ \text{Truth} \begin{array}{c} \text{cancer} \\ \text{normal} \end{array} & \begin{pmatrix} 0 & 1000 \\ 1 & 0 \end{pmatrix} \end{array}
$$

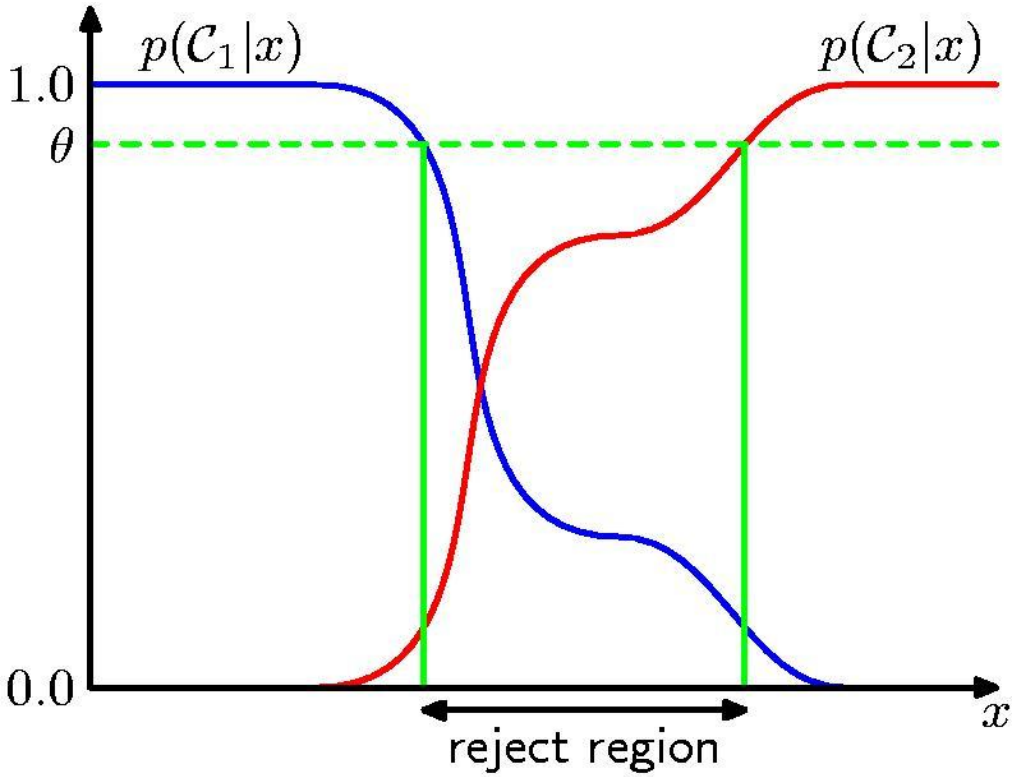
Minimum Expected Loss

$$\mathbb{E}[L] = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj} p(\mathbf{x}, \mathcal{C}_k) \, \mathrm{d}\mathbf{x}$$

Regions $\mathcal{R}_j$ are chosen to minimize

$$\mathbb{E}[L] = \sum_k L_{kj} p(\mathcal{C}_k|\mathbf{x})$$

Reject Option



Why Separate Inference and Decision?

- Minimizing risk (loss matrix may change over time)
- Reject option
- Unbalanced class priors
- Combining models

Decision Theory for Regression

Inference step

Determine $p(\mathbf{x}, t)$

Decision step

For given x, make optimal prediction, y(x), for t.

Loss function: $\mathbb{E}[L] = \iint L(t, y(\mathbf{x}))p(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t$

The Squared Loss Function

$$\mathbb{E}[L] = \iint \{y(\mathbf{x}) - t\}^2 p(\mathbf{x}, t)\, \mathrm{d}\mathbf{x}\, \mathrm{d}t$$

$$\begin{aligned}\{y(\mathbf{x}) - t\}^2 &= \{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}] + \mathbb{E}[t|\mathbf{x}] - t\}^2 \\ &= \{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\}^2 + 2\{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\}\{\mathbb{E}[t|\mathbf{x}] - t\} + \{\mathbb{E}[t|\mathbf{x}] - t\}^2\end{aligned}$$

$$\mathbb{E}[L] = \int \{y(\mathbf{x}) - \mathbb{E}[t|\mathbf{x}]\}^2 p(\mathbf{x})\, \mathrm{d}\mathbf{x} + \int \mathrm{var}\,[t|\mathbf{x}]\, p(\mathbf{x})\, \mathrm{d}\mathbf{x}$$

$$y(\mathbf{x}) = \mathbb{E}[t|\mathbf{x}]$$

Performance measurement

- **How do we know that $h \approx f$?**
  1. Use theorems of computational/statistical learning theory
  2. Try *h* on a new test set of examples
     (use same distribution over example space as training set)

Learning curve = % correct on test set as a function of training set size