Outline

- Schema theorem
- Learning sets of rules

Schemas

How to characterize evolution of population in GA?

Schema = string containing 0, 1, * ("don't care")

- Typical schema: 10**0*
- Instances of above schema: 101101, 100000, ...

Characterize population by number of instances representing each possible schema

- $m(s,t)$ = number of instances of schema s in pop at time t

Consider Just Selection

- $\bar{f}(t)$ = average fitness of pop. at time t
- $m(s,t)$ = instances of schema s in pop at time t
- $\hat{u}(s,t)$ = ave. fitness of instances of s at time t

Probability of selecting h in one selection step

$$\begin{aligned}\Pr(h) &= \frac{f(h)}{\sum_{i=1}^{n} f(h_i)} \\ &= \frac{f(h)}{n\bar{f}(t)}\end{aligned}$$

Probabilty of selecting an instance of s in one step

$$\begin{aligned}\Pr(h \in s) &= \sum_{h \in s \cap p_t} \frac{f(h)}{n\bar{f}(t)} \\ &= \frac{\hat{u}(s,t)}{n\bar{f}(t)} m(s,t)\end{aligned}$$

Expected number of instances of s in n steps

$$E[m(s,t+1)] = \frac{\hat{u}(s,t)}{\bar{f}(t)} m(s,t)$$

Schema Theorem

$$E[m(s,t{+}1)] \geq \frac{\hat{u}(s,t)}{\bar{f}(t)} m(s,t) \left(1 - p_c \frac{d(s)}{l-1}\right) (1{-}p_m)^{o(s)}$$

- $m(s,t)$ = instances of schema s in pop at time t
- $\bar{f}(t)$ = average fitness of pop. at time t
- $\hat{u}(s,t)$ = ave. fitness of instances of s at time t
- p_c = probability of single point crossover operator
- p_m = probability of mutation operator
- l = length of single bit strings
- $o(s)$ number of defined (non "*") bits in s
- $d(s)$ = distance between leftmost, rightmost defined bits in s

Learning Disjunctive Sets of Rules

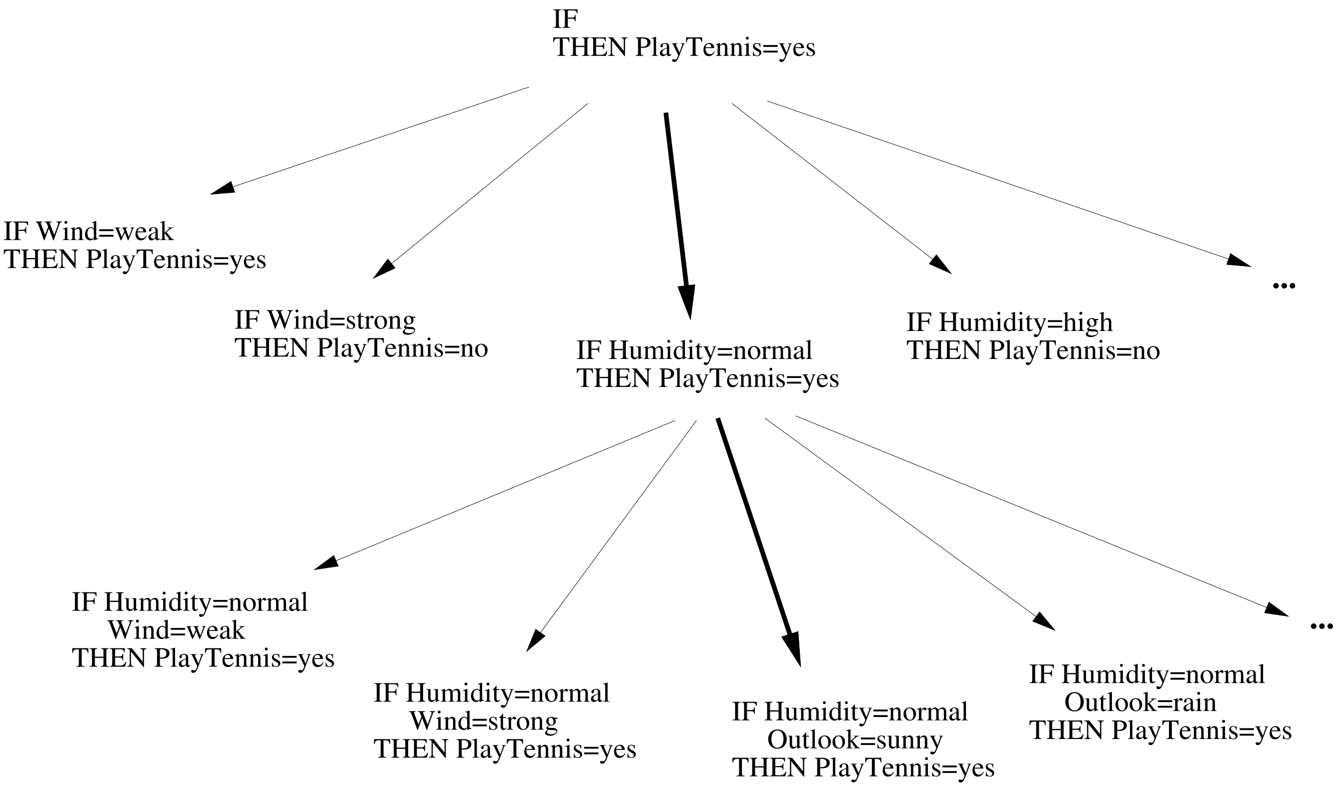
Learn one rule at a time.

- Rules cover positive examples,
- assume uncovered examples are negative (negation as failure)

Sequential covering algorithm:

1. *Learn one rule* with high accuracy, any coverage
2. Remove positive examples covered by rule
3. Repeat

Learn-One-Rule



What evaluation function?
Beam search?

Variants in Searching H

- *Simultaneous* or *sequential* covering of data?
- General $\rightarrow$ specific, or specific $\rightarrow$ general?
- Generate-and-test, or example-driven?
- What statistical evaluation function?

Emergency C-Section Study

- Probability of emergency c-section is 0.12 over entire population of 10,000 patients
- Each patient described by 215 attributes
- Cost-benefit analysis suggests if probability of emergency c-sec > 0.4, then elective c-section improves the expected outcome.

Typical learned rule:

```
If   No previous vaginal deliveries, and
     Abnormal 2nd Trimester Ultrasound, and
     Malpresentation at admission
Then Probability of Emergency C-Section is 0.6

 Training set: 26/41 = .634
 Test set: 12/20 = .600
```

Learning First-order Rules

Why do that?

FOIL($Target\_predicate, Predicates, Examples$)

- $Pos \leftarrow$ positive $Examples$
- $Neg \leftarrow$ negative $Examples$
- while Pos, do

  Learn a NewRule

  – $NewRule \leftarrow$ most general rule possible

  – $NewRuleNeg \leftarrow Neg$

  – while $NewRuleNeg$, do

    Add a new literal to specialize NewRule

    1. $Candidate\_literals \leftarrow$ generate candidates
    2. $Best\_literal \leftarrow$
       $\text{argmax}_{L \in Candidate\_literals} Foil\_Gain(L, NewRule)$
    3. add $Best\_literal$ to $NewRule$ preconditions
    4. $NewRuleNeg \leftarrow$ subset of $NewRuleNeg$
       that satisfies $NewRule$ preconditions

  – $Learned\_rules \leftarrow Learned\_rules + NewRule$

  – $Pos \leftarrow Pos - \{$members of Pos covered by
    $NewRule\}$

- Return $Learned\_rules$

Specializing Rules in FOIL

Learning rule: $P(x_1, x_2, \ldots, x_k) \leftarrow L_1 \ldots L_n$
Specialize by adding new literal of form:

- $Q(v_1, \ldots, v_r)$, where at least one of the v_i in the created literal must already exist as a variable in the rule.
- $Equal(x_j, x_k)$, where x_j and x_k are variables already present in the rule
- The negation of either of the above forms of literals