Session 2

- Rule learning: use IF-THEN rule set to represent concept.
  IF Sky = sunny $\wedge$ Wind = weak THEN $\oplus$
  $\ldots$
  - Separate-and-conquer algorithm
  - Genetic algorithm
- Instance based learning

Separate-and-conquer Rule Learning

```
function LearnRuleSet(E⊕, E⊖):
    LearnedRules := ∅
    while E⊕ ≠ ∅, do
        pick e from E⊕
        Rule := LearnOneRule(e, E⊕, E⊖)
        LearnedRules := LearnedRules ∪ {Rule}
        E⊕ := E⊕ -
            {examples classified correctly by Rule}
    return LearnedRules

function LearnOneRule(e, E⊕, E⊖):
    NewRule := "IF true THEN ⊕"
    NewRuleNeg := E⊖
    while NewRuleNeg ≠ ∅, do
        Candidates := GenerateCandidateLiterals(e)
        BestLit := argmax_{L∈Candidates}
            performance(Specialise(NewRule,L), E⊕, E⊖)
        NewRule := Specialise(NewRule, BestLit)
        NewRuleNeg := {x ∈ E⊖ | x covered by NewRule}
    return NewRule

function Specialise(Rule, Lit):
    let Rule = "IF conditions THEN ⊕"
    return "IF conditions ∧ Lit THEN ⊕"
```

Genetic Algorithms

- Population of individuals each represented as a bit-string.
- Evolution by means of genetic operators
  - Mutation

1	0	0	1	1	0	0	1	1	→	1	0	1	1	1	0	0	1	1

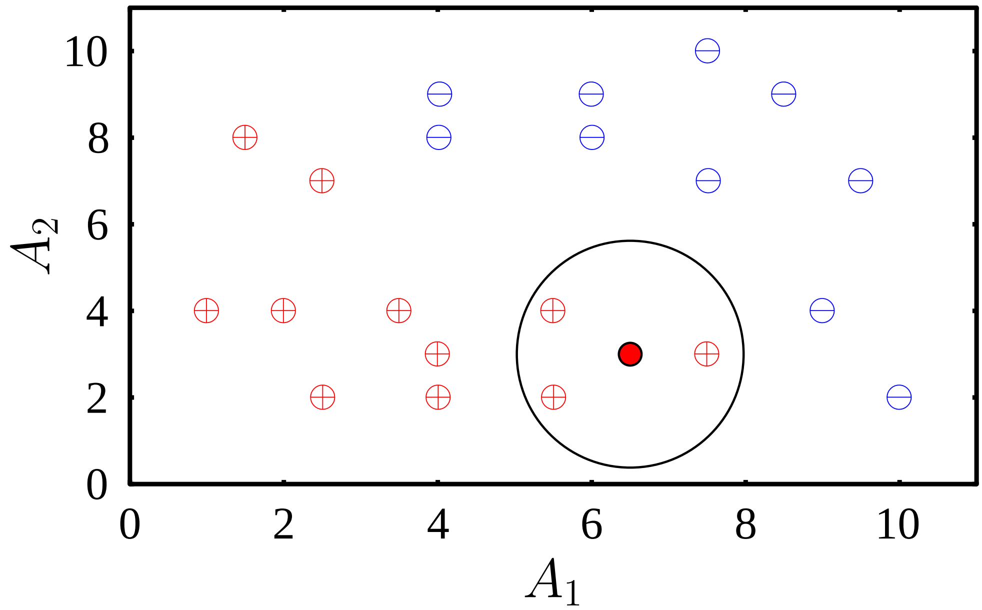
  - Selection
    Select individuals according to fitness
  - Cross-over

1	0	0	1	1	0	0	1	1		1	0	0	0	0	0	0	1	1
									→									
1	1	1	0	0	0	1	1	0		1	1	1	1	1	0	1	1	0

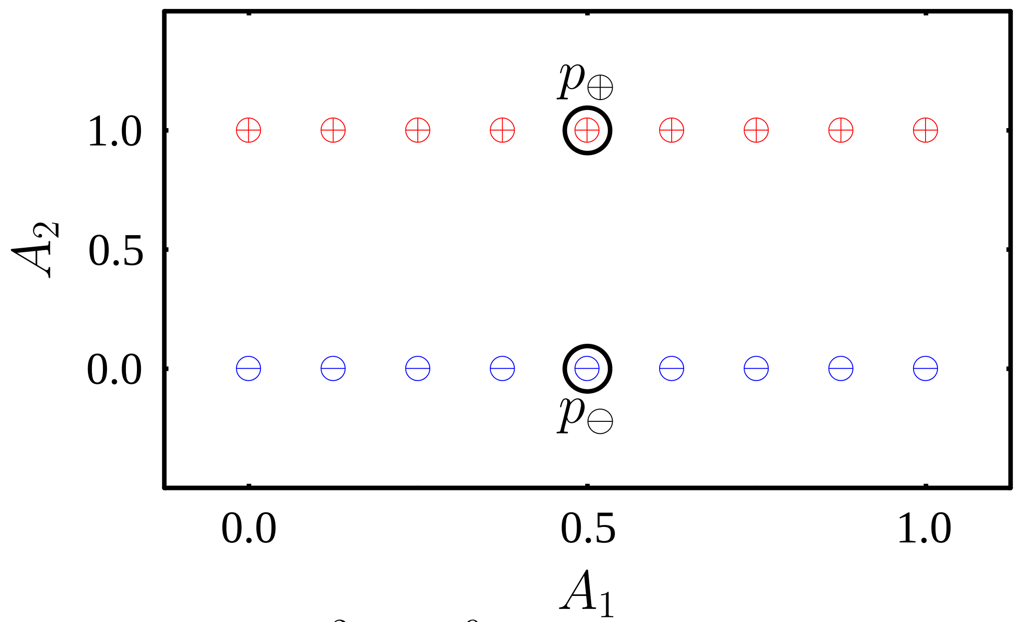
- How to represent rule set as bit-string (GABIL)?
  - 1 bit for each attribute value
  - E.g., conditions involving the attribute "Wind":
    "Wind = strong" → 10
    "Wind = weak" → 01
    "Wind = strong ∨ Wind = weak" → 11
  - When class values are mutually exclusive, use minimal encoding (E.g. one bit for binary problems)
  - The string for a rule set is concatenation of individual rule strings

Instance Based Learning

- kNN: class of new instance = majority vote among the classes of its k neighbors
- Neighbors? $\rightarrow$ distance measure (e.g., Euclidian distance)



- $W(A_i) = 1 - \frac{1}{n_{examples}} \sum_{c=1}^{n_{classes}} \sum_{j=1}^{n_c} d'(p_{ci}, x_{ji})$



$$W(A_2) = 1 - \frac{1}{n} \sum_{c=1}^{2} \sum_{j=1}^{9} 0 = 1$$

$$W(A_1) = 1 - \frac{1}{18} \cdot \underbrace{2}_{\text{classes}} \cdot \underbrace{2 \cdot (0.125 + 0.25 + 0.375 + 0.5)}_{\text{4 symmetric points left and right of } p_c} = 0.72$$