RULE LEARNING

- concept: rules & rule sets 分类问题

  ordered rule set: **a decision list**

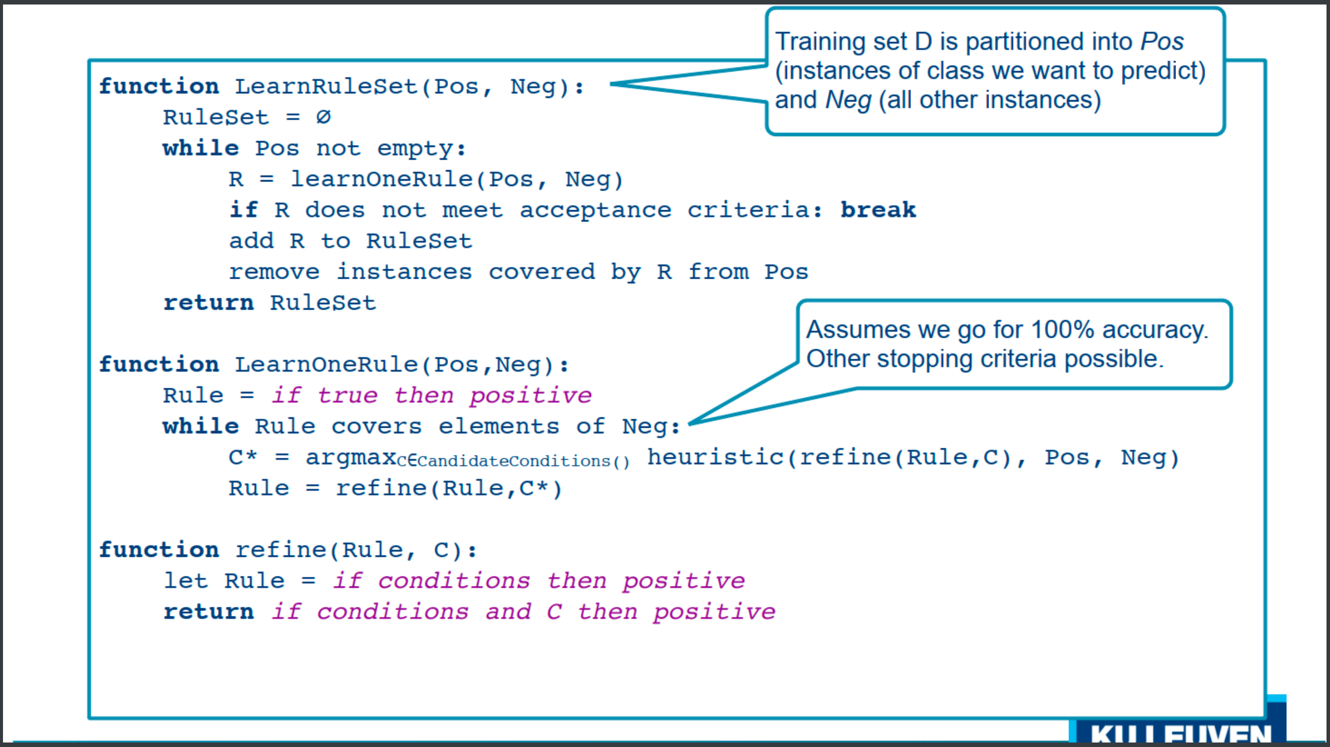
  (rule i only applies if rules 1 to i-1 did not apply, "if - then - else if")

  decision tree -> a set of rules

- sequential covering
  - "separate-and-conquer"：

    每次separate一部分数据满足当前规则，再conquer剩下的数据

  - accuracy & coverage
  - general 算法：



```
function LearnRuleSet(Pos, Neg):
    RuleSet = ∅
    while Pos not empty:
        R = learnOneRule(Pos, Neg)
        if R does not meet acceptance criteria: break
        add R to RuleSet
        remove instances covered by R from Pos
    return RuleSet

function LearnOneRule(Pos,Neg):
    Rule = if true then positive
    while Rule covers elements of Neg:
        C* = $\text{argmax}_{C \in \text{CandidateConditions()}}$ heuristic(refine(Rule,C), Pos, Neg)
        Rule = refine(Rule,C*)

function refine(Rule, C):
    let Rule = if conditions then positive
    return if conditions and C then positive
```

Training set D is partitioned into *Pos* (instances of class we want to predict) and *Neg* (all other instances)

Assumes we go for 100% accuracy. Other stopping criteria possible.

- learning one rule: typically, greedy search

  top-down:

  start with maximally general rule, add conditions

  bottom-up:

  start with maximally specific rule, remove conditions

- heuristic for selecting rules?

goal: high accuracy & reasonably high coverage

accuracy recall the cocktail example

accuracy 值相同的情况下，分母更大对应的rule更好（coverage）

m-estimate

$$\text{m-estimate}(m, q) = \frac{p + mq}{p + n + m}$$

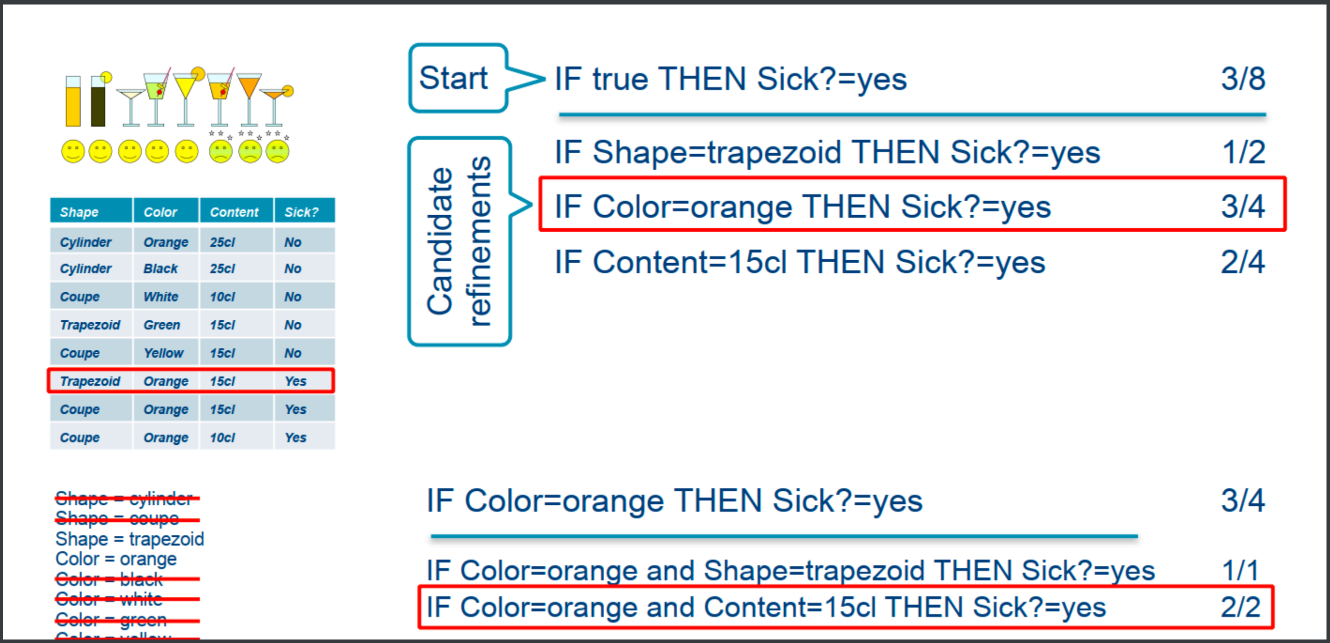
q is a prior estimate of accuracy, *m* is its weight

- when *m* is large, estimate is coser to *q* (more conservative)
- converges to accuracy as *p+n* grows

- 优化 1： example-driven

Idea: pick a not-yet-covered example & use as hypothesis space

e.g.



Shape	Color	Content	Sick?
Cylinder	Orange	25cl	No
Cylinder	Black	25cl	No
Coupe	White	10cl	No
Trapezoid	Green	15cl	No
Coupe	Yellow	15cl	No
Trapezoid	Orange	15cl	Yes
Coupe	Orange	15cl	Yes
Coupe	Orange	10cl	Yes

~~Shape = cylinder~~
~~Shape = coupe~~
Shape = trapezoid
Color = orange
~~Color = black~~
~~Color = white~~
~~Color = green~~
~~Color = yellow~~

Start: IF true THEN Sick?=yes 3/8

Candidate refinements:

IF Shape=trapezoid THEN Sick?=yes 1/2

IF Color=orange THEN Sick?=yes 3/4

IF Content=15cl THEN Sick?=yes 2/4

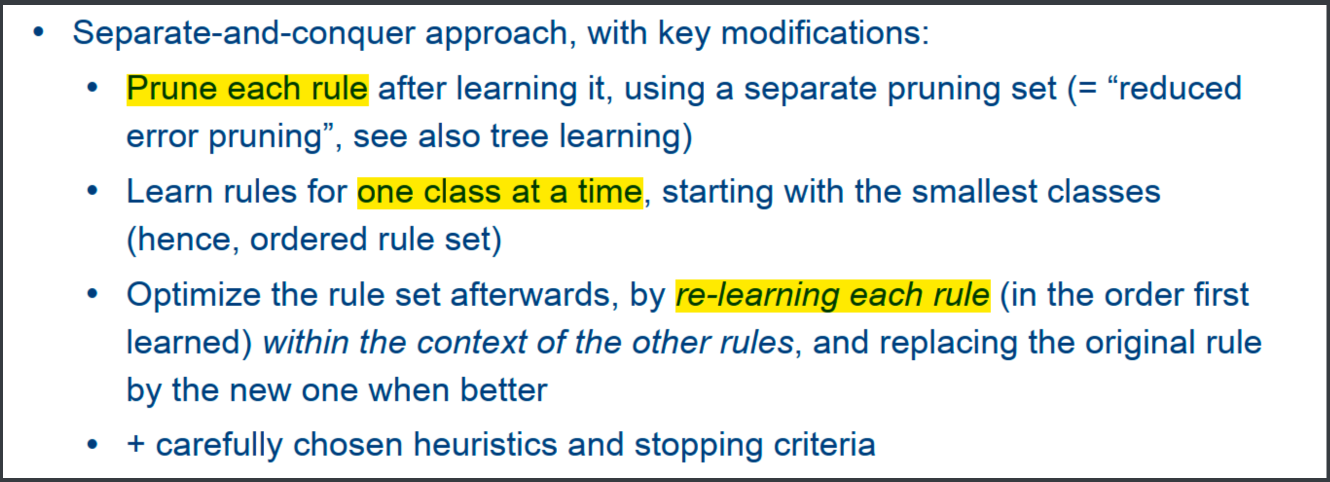
IF Color=orange THEN Sick?=yes 3/4

IF Color=orange and Shape=trapezoid THEN Sick?=yes 1/1

IF Color=orange and Content=15cl THEN Sick?=yes 2/2

被圈出的example只有三个条件，以此作为搜索空间进行第一次搜索；

- 优化 2：RIPPER

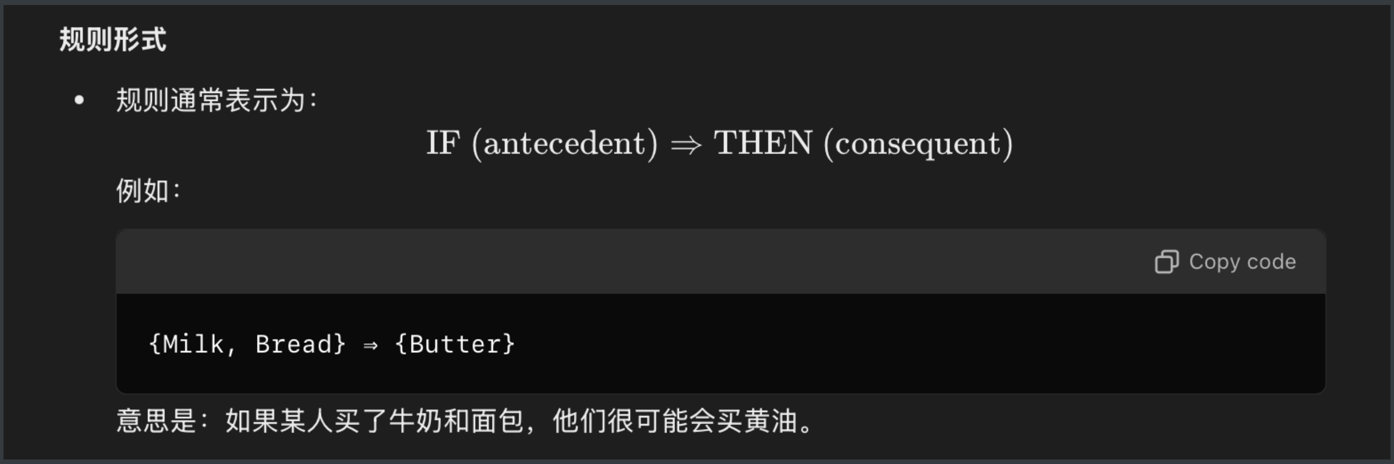


- Separate-and-conquer approach, with key modifications:
  - Prune each rule after learning it, using a separate pruning set (= “reduced error pruning”, see also tree learning)
  - Learn rules for one class at a time, starting with the smallest classes (hence, ordered rule set)
  - Optimize the rule set afterwards, by *re-learning each rule* (in the order first learned) *within the context of the other rules*, and replacing the original rule by the new one when better
  - + carefully chosen heuristics and stopping criteria

association rules

vs. classification rules:

association rules are descriptive rules **indicating patterns in data**;



规则形式

- 规则通常表示为：

$$\text{IF (antecedent)} \Rightarrow \text{THEN (consequent)}$$

例如：

```
{Milk, Bread} ⇒ {Butter}
```

意思是：如果某人买了牛奶和面包，他们很可能会买黄油。