

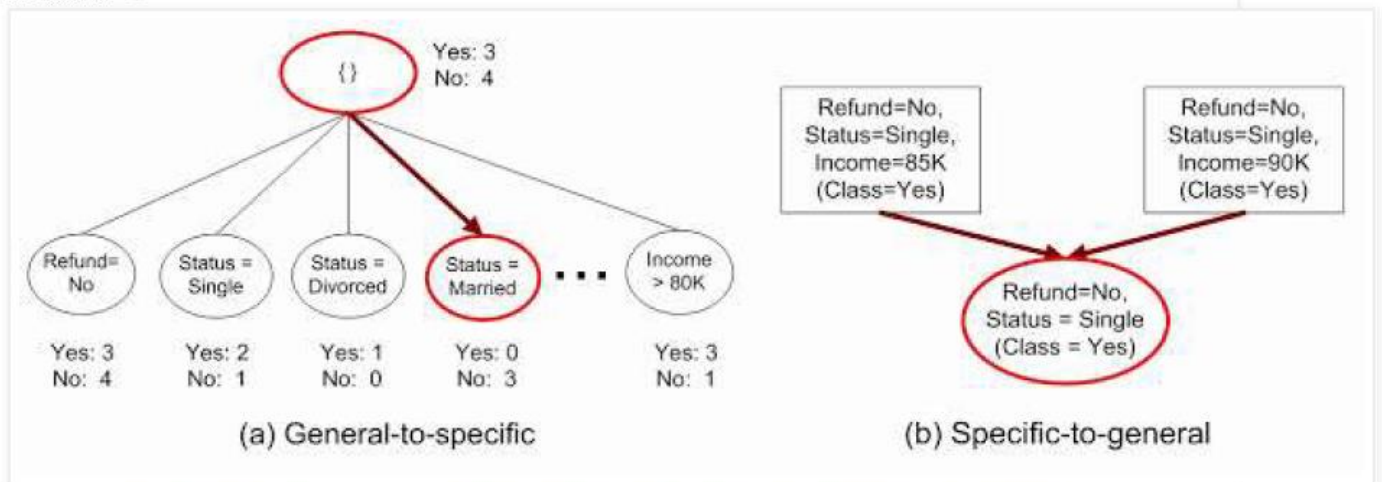
4. Rule-Based Classifier 規則分類直接方式的連續覆蓋 (Direct Method: Sequential Covering)

接著連續覆蓋的簡介共有五個：

- 1.規則的生長 (Rule Growing)
- 2.記錄的削減 (Instance Elimination)
- 3.規則的評估 (Rule Evaluation)
- 4.生長停止的標準 (Stopping Criterion)
- 5.規則的修剪 (Rule Pruning)

簡介1・規則的生長 (Rule Growing)

規則的生長可以分成演繹法 (general-to-specific) 與歸納法 (specific-to-general) 兩種，這用圖比較好解釋。



而其中，演繹法是採用一個空集合，在慢慢改善選取目標。歸納法則是隨機挑選一個目標Class的資料作為基礎，在慢慢修正

補充：

CN2演算法的步驟是

- 1.先建立一個空的規則， $r:\{\} \rightarrow y$
- 2.開始加入規則條件進入，挑選最小Entropy的條件
- 3.確定規則後，挑選能覆蓋最多目標Class的規則

PIPPER演算法的步驟是 (之後會在詳細介紹)

- 1.同樣建立一個空的規則， $r:\{\} \rightarrow y$
- 2.開始加入規則條件進入，挑選最大FOIL's Information Gain的。

$R0: \{\} \Rightarrow \text{class}$ (initial rule)

$R1: \{A\} \Rightarrow \text{class}$ (rule after adding conjunct)

$\text{Gain}(R0, R1) = t [\log(p1/(p1+n1)) - \log(p0/(p0 + n0))]$

where t : number of positive instances covered by both $R0$ and $R1$

$p0$: number of positive instances covered by $R0$

$n0$: number of negative instances covered by $R0$

$p1$: number of positive instances covered by $R1$

$n1$: number of negative instances covered by $R1$

簡介2 · 記錄的削減 (Instance Elimination)

1. 在做連續覆蓋時為什麼要把記錄給消除掉呢？

因為如果不刪除的話，下一個規則會重複覆蓋到。

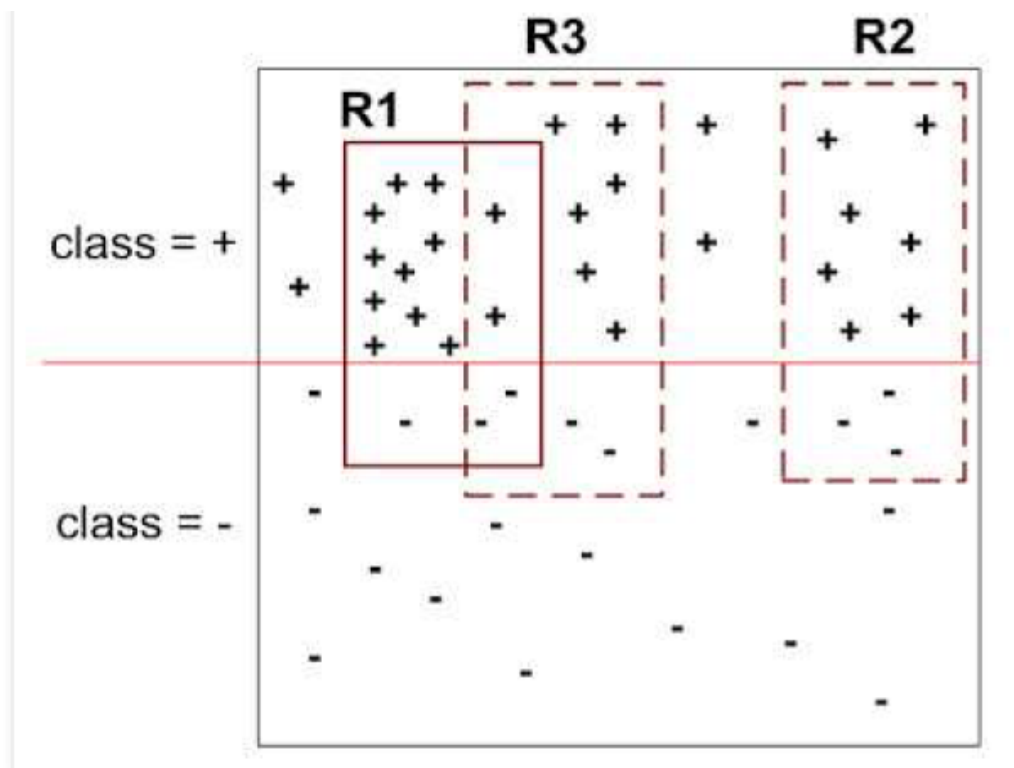
2. 為什麼要刪除目標Class (Positive) 的記錄？

確保下一個規則是不同一個的。(還是跟上面理由差不多)

3. 為什麼也要刪除不是目標Class (Negative) 的記錄？

防止規則被低估了精確性。

如下圖所式：



上頭共有，29個 + 與 21個 -，我們選出了三個規則，

R1，初始正確率：12/15 (80%)

R2，初始正確率：7/10 (70%)

R3，初始正確率：8/12 (66.7%)

很清楚看到R1是最好的策略，必須將R1所覆蓋的資料全部移除，進而繼續挑選下一個規則。

接著，R3因為R1資料的移除，故R3的正確率就變成了6/8 (75%)，高於R2，故我們選R3。

也就是說，如果我們不刪除R1的覆蓋的資料，R3就會重複覆蓋R1的也覆蓋的資料。另外如果不移除R1所覆蓋的+，我們可能會高估了R3的正確性，或者我們不移除R1覆蓋的-，也可能會低估了R3的正確性。

簡介3 · 規則的評估 (Rule Evaluation)

規則的評估，當然還是用計量的方式，總共有以下幾種

1. 正確性 (Accuracy) :

$$\text{Accuracy} = \frac{n_c}{n}$$

2. Laplace法 :

$$\text{Laplace} = \frac{n_c + 1}{n + k}$$

3. M-estimate法 :

$$\text{M-estimate} = \frac{n_c + kp}{n + k}$$

其中，

N 是規則能夠覆蓋多少記錄

Nc 是規則能夠覆蓋多少目標Class記錄

k 是總共有多少的組數

p 是事前機率 (Prior probability)

簡介4與5 · 生長停止的標準 (Stopping Criterion) 與 規則的修剪 (Rule Pruning)

停止生長的標準在於Gain值有沒有改善，如果沒有改善就停止

而規則的修剪，很像決策樹的減少錯誤修剪 (回顧 [Reduced Error Pruning](#))，步驟很像

1. 先移除一個規則中的條件

2. 套入驗證資料計算移除前與移除後的比率

3. 如果移除後有改善則移除，則修剪這個規則