

Searching for the Optimal Ordering of Classes in Rule Induction

Sezin Ata Olcay Taner Yıldız

Department of Computer Engineering, Işık University, TR-34980, Istanbul, Turkey

ICPR 2012

Outline

- 1 Introduction
- 2 Proposed Algorithms
 - Forward Ordering Search
 - Pairwise Error Approximation
- 3 Experiments
- 4 Conclusions

Rule Induction

- A rule set is typically an ordered list of rules.
- A rule contains a conjunction of terms and a class code.
- The terms are of the form $x_i = v$, $x_i < \theta$ or $x_i \geq \theta$.
- Example rule (Iris dataset)

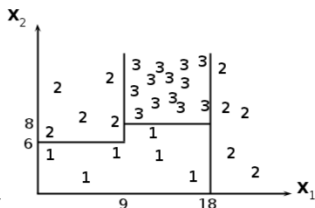
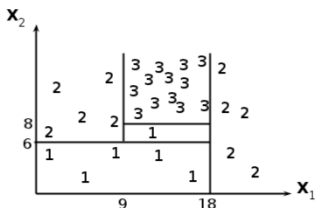
If $F_3 < 1.9$ **and** $F_4 \geq 5.1$ **Then** iris-setosa

Else

If $F_3 < 4.7$ **Then** iris-versicolor

Else iris-virginica

Motivation



- a) Current Approach' s Rule Set
- If $(x_1 < 18) \&\& (x_2 < 6)$
 Then Class = 1
 - If $(x_1 > 9) \&\& (x_2 < 8)$
 Then Class = 1
 - If $(x_1 < 9)$
 Then Class = 2
 - If $(x_1 > 18)$
 Then Class = 2
 - Else
 Class = 3
- b) Another Ordering' s Rule Set
- If $(x_1 < 9) \&\& (x_2 > 6)$
 Then Class = 2
 - If $(x_1 > 18)$
 Then Class = 2
 - If $(x_2 < 8)$
 Then Class = 1
 - Else
 Class = 3

Ripper

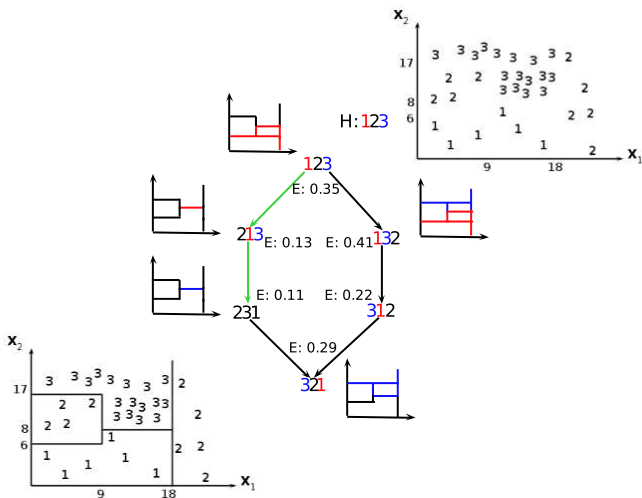
```

1  RS = {}
2  Ci ordered in increasing prior probability
3  for p = 1 to K - 1
4      Pos = Cp, Neg = Cp+1, ..., CK
5      RSp = {}
6      while D contains positive samples
7          Divide D into Grow set G and Prune set P
8          r = GrowRule(G)
9          PruneRule(r, P)
10         if CalculateError(r) > 0.5
11             break
12         else
13             RSp = RSp + r
14             Remove examples covered by r from D
15         for i = 1 to 2
16             OptimizeRuleset(RSp, D)
17             SimplifyRuleset(RSp, D)
18         RS = RS + RSp
19     return RS
    
```

Forward Ordering Search: Algorithm

- Views optimizing the ordering of classes in Ripper as a search in the state space of all possible orderings.
- Forward search algorithm starts from an initial state (heuristic ordering) and use exchange operators to generate candidate states (orderings).
- Let say we have the ordering $C_1 C_2 C_3 \dots C_{K-1} C_K$. The exchange operator creates the following $K - 1$ candidate orderings: $C_2 C_1 C_3 \dots C_{K-1} C_K$, $C_1 C_3 C_2 \dots C_{K-1} C_K$, $C_1 C_2 C_4 \dots C_{K-1} C_K$, \dots , $C_1 C_2 C_3 \dots C_K C_{K-1}$.
- Candidates are evaluated via 10×10 -fold cross-validation.
- We stop the search when no candidate improves on the current best.

Forward Ordering Search: Example

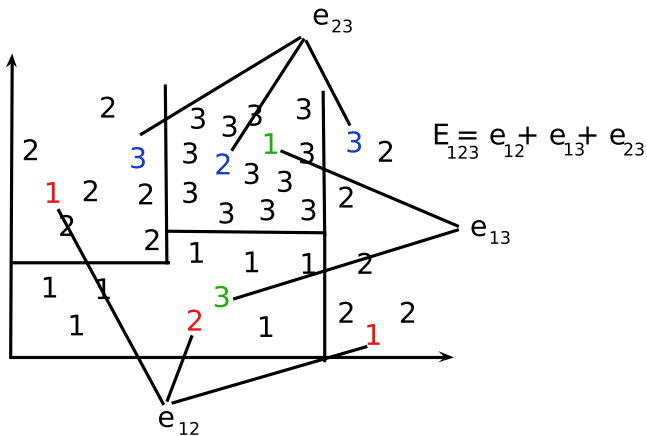


Pairwise Error Approximation: Assumption

The expected error of the Ripper trained with the ordering π_i , is the sum of $K(K - 1)/2$ pairwise expected errors of classes:

$$\hat{E}_{\pi_i} = \sum_{j=1}^K \sum_{k=1, [j] <_{\pi_i} [k]}^K e_{jk}$$

Pairwise Error Approximation: Assumption



Pairwise Error Approximation: Algorithm

- Run Ripper algorithm N times with N random orderings π_i and get the test errors E_{π_i} .
- Minimize the total estimation error

$$E_t = \sum_{i=1}^N (E_{\pi_i} - \hat{E}_{\pi_i})^2$$

- Solve the following system of linear equations

$$\forall_{j,k} \frac{\partial E_t}{\partial e_{jk}} = 0$$

- Solved e_{jk} 's, search all possible class orderings to get the optimal ordering.

Datasets

Dataset	# of attributes	# of classes	Sample size
<i>led7</i>	7	7	3200
<i>ocr</i>	256	10	600
<i>optdigits</i>	64	10	3823
<i>pendigits</i>	16	10	7494
<i>segment</i>	19	7	2310
<i>shuttle</i>	9	7	58000
<i>winequality</i>	11	7	6497
<i>yeast</i>	8	10	1484

Results: Error rates

Dataset	Ripper	FOS	PEA
led7	31.83 ± 0.23	29.67 ± 0.23	28.93 ± 0.21
ocr	26.61 ± 0.58	24.73 ± 0.59	22.08 ± 0.54
opt	10.96 ± 0.14	10.55 ± 0.15	8.57 ± 0.12
pen	5.32 ± 0.07	4.87 ± 0.08	4.49 ± 0.08
seg	6.54 ± 0.17	4.38 ± 0.12	5.03 ± 0.14
shu	0.04 ± 0.00	0.03 ± 0.00	0.01 ± 0.00
wine	46.32 ± 0.16	46.23 ± 0.13	56.64 ± 0.23
yea	43.09 ± 0.35	42.39 ± 0.40	43.38 ± 0.36

Results: Complexity of FOS

Dataset	# of Orderings
led7	40
ocr	18
opt	25
pen	41
seg	22
shu	12
wine	17
yea	25

Results: Average Estimation Error of PEA

Dataset	\bar{E}_t
led7	0.31
ocr	0.63
opt	0.20
pen	0.08
seg	0.24
shu	1.86
wine	3.41
yea	2.67

Summary

- Current heuristic approach that orders the classes in a dataset according to their sample sizes, usually does not give the most accurate classification.
- FOS starts with the ordering the heuristic provides and searches for better orderings by swapping consecutive classes.
- PEA transforms the ordering search problem into an optimization problem and uses the solution of the optimization problem to extract the optimal ordering.