Incremental Construction of Rule Ensembles using Classifiers Produced by Different Class Orderings

Olcay Taner Yıldız[1], Aydın Ulaş[2]

[1]Department of Computer Engineering, Işık University, TR-34980, Istanbul, Turkey
[2]ARGELA A.Ş., Network & Services Innovation Center, Istanbul, Turkey

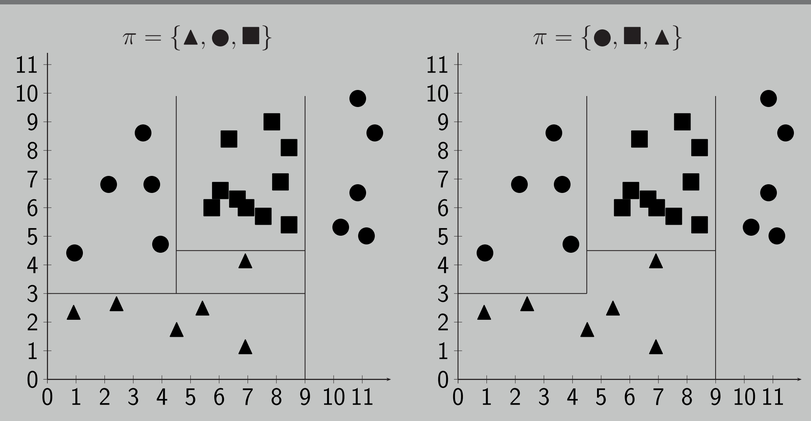
Rule Induction

- 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 $x_i \geq \theta$.

Ripper

```
1  Ruleset Ripper(D, π)
2    RS = {}
3    for p = 1 to K − 1
4      Pos = π_p, Neg = π_{p+1}, ..., π_K
5      RS_p = {}
6      while D contains positive samples do
7        Divide D into Grow set G and Prune set P
8        r = GrowRule(G)
9        PruneRule(r, P)
10       RS_p = RS_p + r
11       Remove examples covered by r from D
12     for i = 1 to 2
13       OptimizeRuleset(RS_p, D)
14     RS = RS + RS_p
15   return RS
```

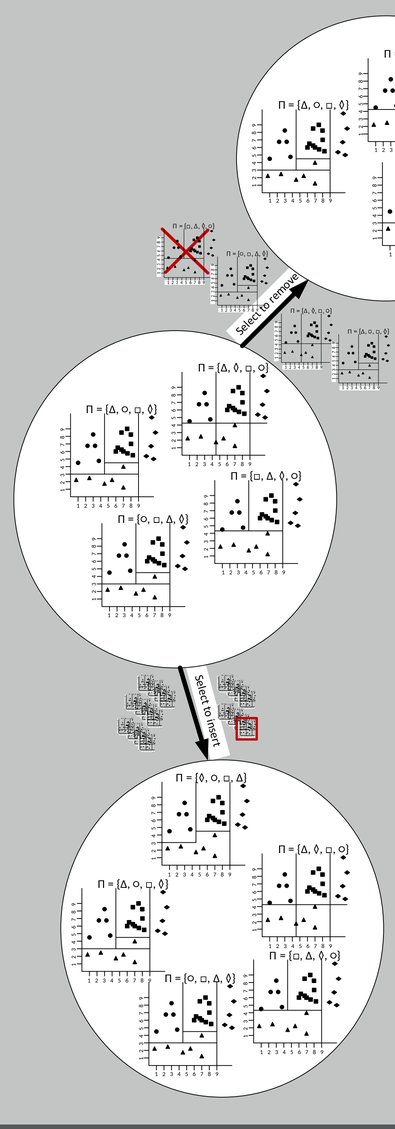
Different Orderings Result in Different Rulesets



If $(x_1 < 9)$ **and** $(x_2 < 3)$
  Then class = ▲
Else
  If $(x_1 > 4.5)$ **and** $(x_2 < 4.5)$
    Then class = ▲
  Else
    If $(x_1 < 4.5)$
      Then class = ●
    Else
      If $(x_1 > 9)$
        Then class = ●
      Else class = ■

If $(x_1 > 9)$
  Then class = ●
Else
  If $(x_1 < 4.5)$ **and** $(x_2 > 3)$
    Then class = ●
  Else
    If $(x_2 > 4.5)$
      Then class = ■
    Else class = ▲

Ren Example



Motivation

- A rule classifier RC_i with class ordering π_i may be accurate for some classes but not all.
- Each rule classifier with a different class ordering has a certain inductive bias.
- Best to use RC_i with other rule classifiers which are good at other classes.

Proposed Approach: Rule Ensemble (Ren)

- Views ensemble construction problem as an optimization problem.
- Search in the state space of all possible ensembles.
- Search space contains all possible ensembles of rule classifiers.

Ren

```
1  Ensemble Ren(D, m)
2    S = {}
3    π = Generate an array of m random permutations
4    for i = 1 to m
5      RS_i = Ripper(D, π_i)
6      S[i].RS = RS_i
7      S[i].π = π_i
8    bestError = Error(S)
9    improved = true
10   while improved
11     improved = false
12     for i = 1 to size(S)
13       π = NeighborPermutations(S[i].π)
14       for j = 1 to K − 1
15         RS_j = Ripper(D, π_j)
16         E = Error(S ∪ (RS_j, π_j))
17         if E < bestError
18           improved = true
19           bestError = E
20           bestCandidate = (RS_j, π_j)
21           bestOperation = Add
22     for i = 1 to size(S)
23       E = Error(S − (S[i].RS, S[i].π))
24       if E < bestError
25         improved = true
26         bestError = E
27         bestCandidate = (S[i].RS, S[i].π)
28         bestOperation = Remove
29     if improved
30       if bestOperation = Add
31         S = S ∪ bestCandidate
32       else
33         S = S − bestCandidate
34   return S
```

Neighborhood function

Let $\pi_0 = C_1 C_2 C_3 \ldots C_{K-1} C_K$. Neighbors of π_0

- $\pi_1 = C_2 C_1 C_3 \ldots C_{K-1} C_K$
- $\pi_2 = C_1 C_3 C_2 \ldots C_{K-1} C_K$
- ...
- $\pi_{K-1} = C_1 C_2 C_3 \ldots C_K C_{K-1}$

Details of the datasets

Dataset	d	C	N
balance	4	3	625
cmc	9	3	1473
hayesroth	4	3	160
iris	4	3	150
leukemia1	5327	3	72
leukemia2	11225	3	72
splice	60	3	3175
tae	5	3	151
wave	21	3	5000
wine	13	3	178
braintumor2	10367	4	50
car	6	4	1728
srbct	2308	4	83
vehicle	18	4	846
braintumor1	5920	5	90
pageblock	10	5	5473
dermatology	34	6	366
segment	19	7	2310
led7	7	10	3200
mfeatmor	6	10	2000
ocr	256	10	600
yeast	8	10	1484

Compared Ensembles

- BEST: We order the base classifiers in terms of accuracy and use the best of them.
- RND: We randomly choose K base classifiers with K different class orderings.
- REN$_m$: The proposed classifier ensemble which uses m base classifiers in the beginning of the floating search.
- ALL: All the available base classifiers are combined without selection.

Average error rates of the algorithms

Dataset	BEST	RND	ALL	REN$_K$	REN$_1$
balance	27.837	28.192	26.106	25.855	26.169
cmc	47.196	47.596	46.870	46.541	46.451
hayesroth	32.184	33.742	30.120	29.374	29.145
iris	5.441	6.900	5.650	5.585	6.130
leukemia1	21.812	23.460	22.572	19.895	20.152
leukemia2	49.819	45.854	44.167	44.167	44.167
splice	6.456	6.498	5.435	5.378	5.378
tae	66.265	66.387	66.469	66.265	66.306
wave	24.400	20.729	18.944	18.950	18.950
wine	8.989	10.298	5.259	4.928	5.945
braintumor2	71.961	71.961	71.961	71.961	71.961
car	9.783	11.976	10.565	8.625	8.512
srbct	34.725	30.398	24.451	24.865	26.514
vehicle	36.441	35.476	32.775	32.162	32.514
braintumor1	31.034	31.241	31.034	30.793	31.103
pageblock	3.565	3.430	3.236	3.118	3.256
dermatology	3.404	4.534	3.405	3.364	5.156
segment	6.286	5.300	4.104	3.940	4.831
led7	29.389	28.017	27.240	26.965	29.179
mfeatmor	30.436	28.385	27.794	27.586	28.408
ocr	32.400	22.270	18.100	18.210	23.660
yeast	43.390	42.542	41.706	41.158	42.297

Average ensemble size of Ren$_K$, Ren$_1$, and All

Dataset	REN$_K$	REN$_1$	ALL
balance	3.4	3.0	6
cmc	2.9	2.0	6
hayesroth	3.6	3.1	6
iris	3.0	2.0	6
leukemia1	4.6	4.2	6
leukemia2	4.1	4.3	6
splice	5.0	5.0	6
tae	2.5	1.4	6
wave	6.0	6.0	6
wine	3.8	3.2	6
braintumor2	4.0	1.0	24
car	4.1	2.7	24
srbct	6.6	5.1	24
vehicle	6.8	7.4	24
braintumor1	5.7	1.7	120
pageblock	8.2	6.3	120
dermatology	9.2	2.9	720
segment	14.9	10.9	1161
led7	18.2	5.0	2075
mfeatmor	14.7	10.2	2151
ocr	18.2	13.5	2542
yeast	15.7	6.4	1983

Average number of states visited by Ren$_K$ and Ren$_1$

Dataset	REN$_K$	REN$_1$
balance	14.0	11.7
cmc	16.6	14.8
hayesroth	14.4	14.3
iris	13.0	7.4
leukemia1	16.4	16.5
leukemia2	11.8	15.3
splice	18.2	18.8
tae	8.7	4.2
wave	20.0	22.7
wine	13.9	12.0
braintumor2	14.5	4.0
car	91.3	43.3
srbct	66.0	45.5
vehicle	91.6	82.3
braintumor1	41.4	9.9
pageblock	149.2	101.1
dermatology	204.0	35.4
segment	701.3	328.4
led7	1614.2	168.3
mfeatmor	786.1	607.6
ocr	1333.2	866.8
yeast	1003.4	270.0

olcaytaner@isikun.edu.tr, aydin.ulas@argela.com.tr