Prioritization of confidence in the associative classifier PGN: strong and weak points

Iliya Mitov[1], Benoit Depaire[2], Krassimira Ivanova[1]

1. Institute of Mathematics and Informatics, Bulgarian Academy of Sciences
2. Hasselt University, Belgium

7-10.07.2014 **Mathematics Days in Sofia**

Associative classifiers

CAR Classifiers - advantages

- very efficient training
- no assumptions on attribute dependence/independence
- very fast classification
- high accuracy
- easily interpreted by humans classification model

CAR Classifiers - examples

- CBA [Liu et al, 1998]
- CMAR [Li et al, 2001]
- ARC-AC and ARC-BC [Zaïane and Antonie, 2002]
- CPAR [Yin and Han, 2003]
- CorClass [Zimmermann and De Raedt, 2004]
- ACRI [Rak et al, 2005]
- TFPC [Coenen and Leng, 2005]
- HARMONY [Wang and Karypis, 2005]
- MCAR [Thabtah et al, 2005]
- CACA [Tang and Liao, 2007]
- ARUBAS [Depaire et al, 2008]

CAR Classifiers - structure

1. **Association rule mining** - typical data mining task that works in an unsupervised manner

2. **Pruning** – to build accurate and compact recognition model

3. **Recognition**

Association Rule Mining

Several techniques for creating association rules are used:

- Apriori algorithm (CBA, ARC-AC, ARC-BC, ACRI, ARUBAS);
- FP-tree algorithm (CMAR);
- FOIL algorithm (CPAR);
- Morishita & Sese Framework (CorClass).

Generating association rules can be made:

- from all training transactions together (CBA,CMAR, ARC-AC)
- for transactions grouped by class label (ARC-BC)

Pruning

- **Pre-pruning / Post-pruning**

- **Isolated pruning techniques** (evaluated individually, in isolation from the other CARs): minimum support , minimum confidence , pessimistic error (CBA)…

- **Non-isolated pruning techniques** (take multiple rules into account when deciding whether or not to prune a specific rule): data coverage (CBA, ARC-AC, ARC-BC and CMAR); correlation between consequent and antecedent (CMAR) …

Recognition

Once the CARs are generated and pruned, the associative classifier uses all these pieces of local knowledge to classify new instances.

Approaches:

- using a single rule (CBA, CorClass, ACRI)
- using a subset of rules (CPAR)
- using all rules (CMAR)

Order-based combined measures for a subset or all rules:

- Select all matching rules
- Group rules per class value
- Order rules per class value according to criterion
- Calculate combined measure for best Z rules
- Laplace Accuracy (CPAR)

P G N

PGN – very short

- The association rule mining goes from the longest rules (instances) to the shorter ones until no intersections between patterns in the classes are possible.
- At the first step of the pruning phase the contradictions of more general rules between classes are cleared.
- After that the pattern set is compacted excluding all more concrete rules within the classes.

PGN - specifics

- Associative classifier
- Small difference – operating over rectangular, but not transactional datasets
- Main difference – prioritizing confidence before the support

Transactional vs Rectangular Data

- Transactional data - Set of items
  - X1 = {A,B,D,E}
  - X2 = {A,C,D}
  - X3 = {B}
- Rectangular/classification data - Set of attribute-value pairs
  - X1 = {A=1, B=1, C=0, D=1, E=1}
  - X2 = {A=1, B=0, C=1, D=1, E=0}
  - X3 = {A=0, B=1, C=0, D=0, E=0}
- Attribute Value pair:
  - Attribute + value
  - Condition
  - c_a = <A,1>

Example dataset

```
R1:  (1| 1, 2, 4, 1)
R2:  (1| 1, 2, 3, 1)
R3:  (1| 3, 1, 3, 2)
R4:  (1| 3, 1, 4, 2)
R5:  (1| 1, 2, 4, 1)      Equal to R1
R6:  (1| 3, 1, 4, 2)      Equal to R4
R7:  (2| 3, 1, 1, 2)
R8:  (2| 2, 1, 1, 2)
R9:  (2| 3, 1, 2, 2)
```

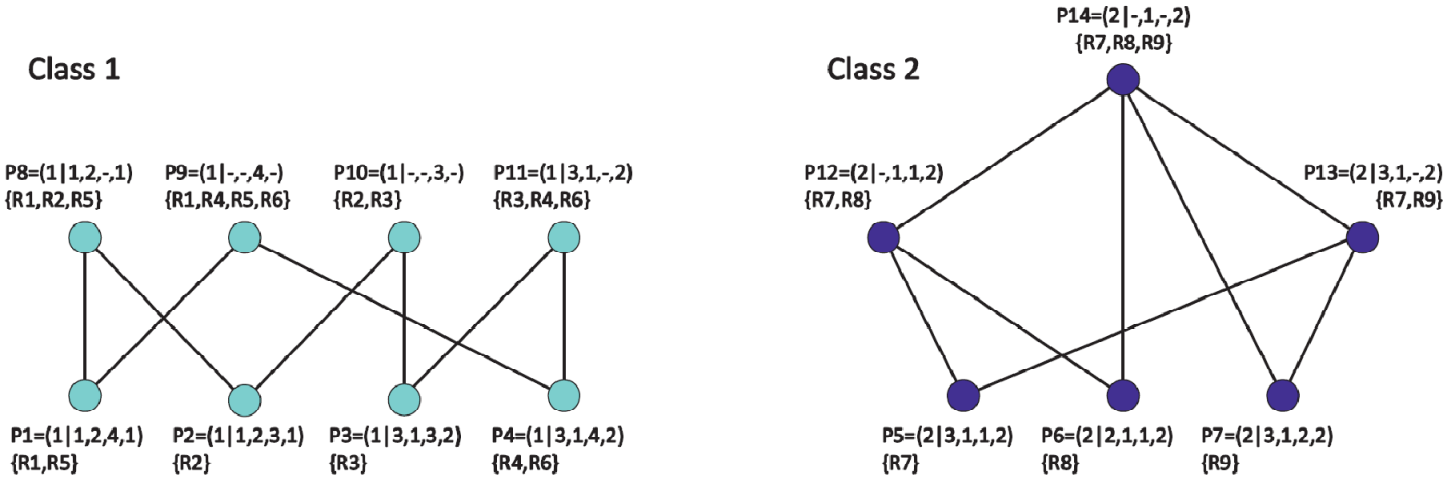
Training – 1. Associative Rule Mining

1. Adding instances to the sub-set in the pattern set, correspondingly to their class-labels.

$$LS = \left\{ R^i \right\} \quad i = 1, \ldots, t$$

Training – 1. Associative Rule Mining

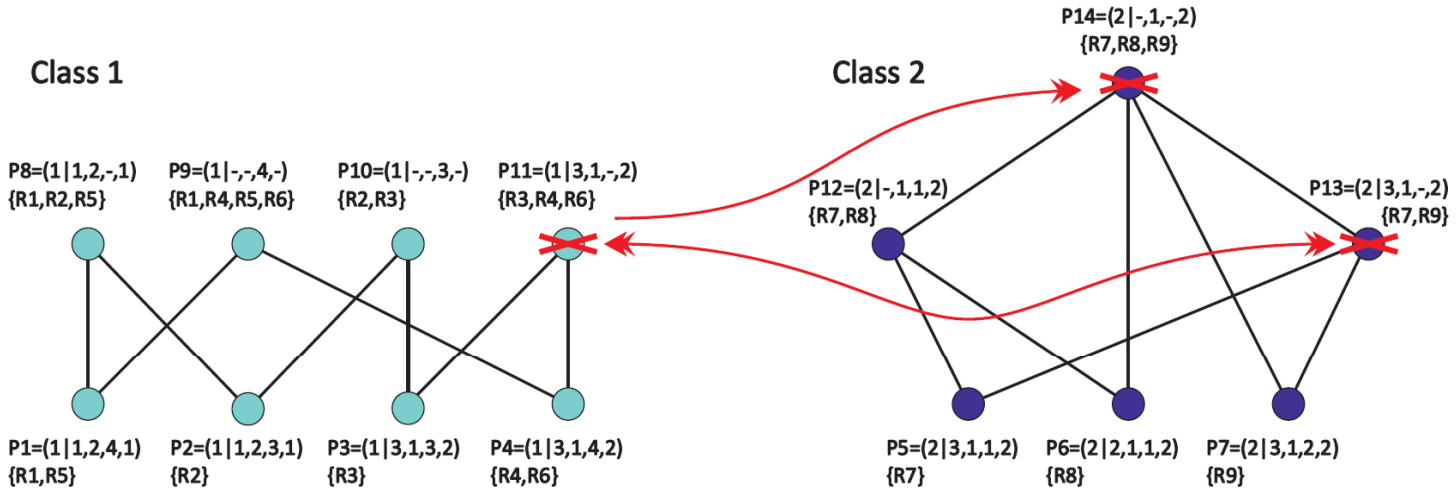
2. Creating intersections between patterns within the class.



$$PS = \left\{ P^l \right\} \qquad P^l : \begin{cases} R^l \in LS \\ P^l = P^i \cap P^j; \;\; P^i \in PS, P^j \in PS, c^i = c^j; \left| P^l \right| > 0 \end{cases}$$

Training – 2. Pruning

(1) Deleting all contradictions and more general with exception patterns in other class

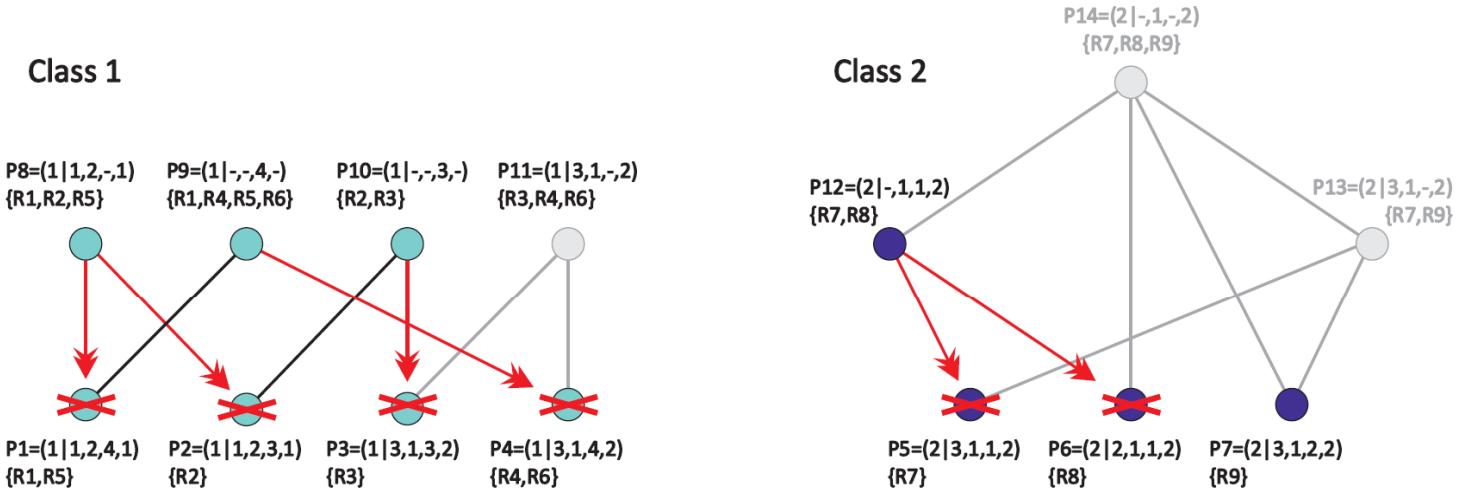


$$P^i, P^j \in PS,\ c^i \neq c^j : \begin{cases} \left|P^i \cap P^j\right| = \left|P^i\right| < \left|P^j\right| : \text{remove } P^i \\ \left|P^i \cap P^j\right| = \left|P^j\right| < \left|P^i\right| : \text{remove } P^j \\ \left|P^i \cap P^j\right| = \left|P^i\right| = \left|P^j\right| : \text{remove } P^i, P^j \end{cases}$$

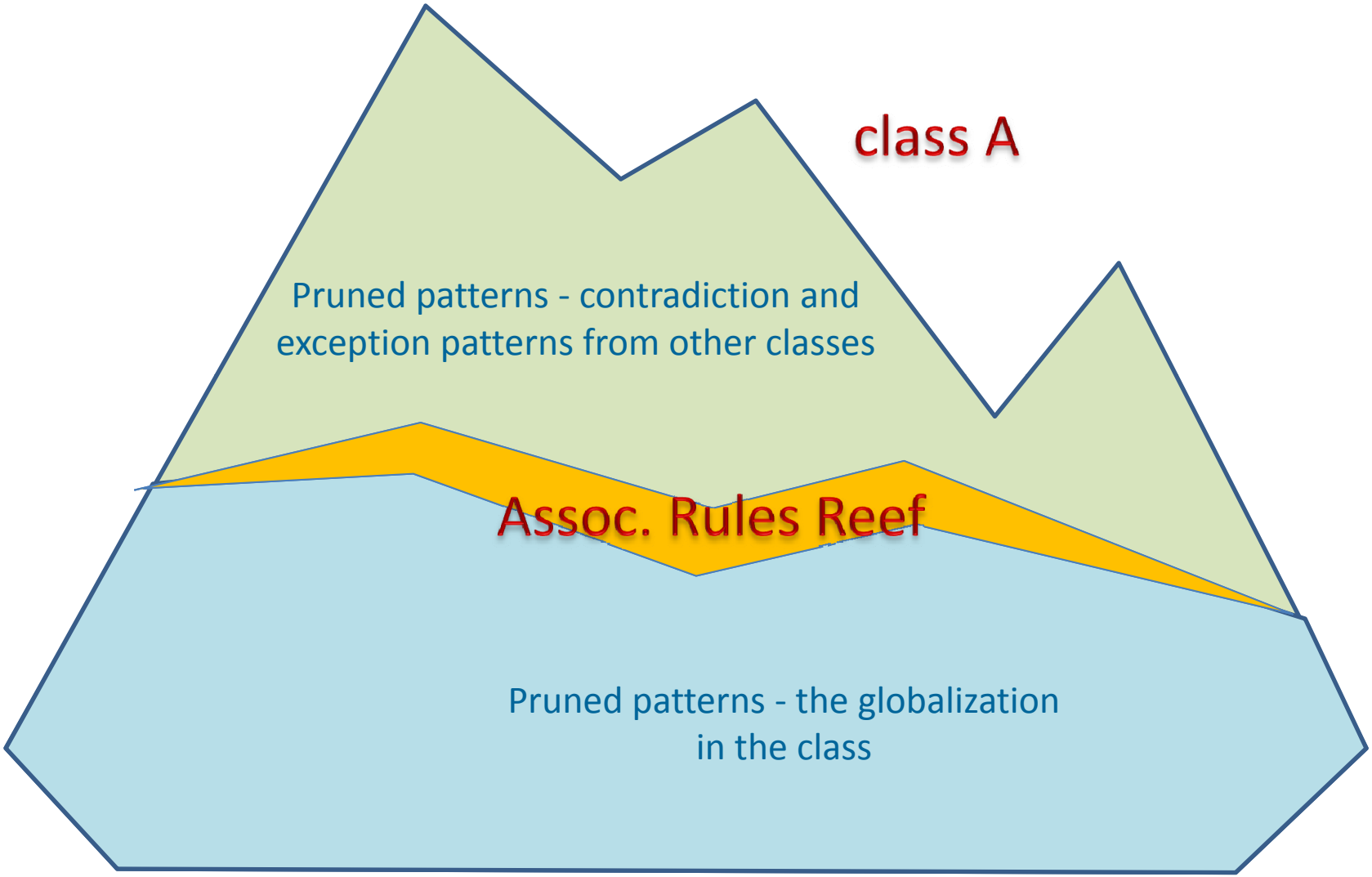
PGN - training process

- **pruning**

  (2) Removing more concrete patterns within the classes.



$$P^i, P^j \in PS,\ c^i = c^j : \begin{cases} \left|P^i \cap P^j\right| = \left|P^i\right| < \left|P^j\right| : \text{remove } P^j \\ \left|P^i \cap P^j\right| = \left|P^j\right| < \left|P^i\right| : \text{remove } P^i \end{cases}$$


class A
Pruned patterns - contradiction and exception patterns from other classes
Assoc. Rules Reef
Pruned patterns - the globalization in the class

PGN - Recognition

Query: $Q = (? \mid a_1, a_2, \ldots, a_n)$

The association rule size corresponds to the number of input attributes which have a non-missing value: $|P| = |\{a_i \mid 1 \le i \le n-1, a_i \neq "-"\}|$

The intersection percentage between pattern P and query Q:

$$IP(P,Q) = 100 * \frac{|P \cap Q|}{|P|}$$

During the recognition stage all patterns from the pattern sets, which have maximal intersection percentage with the query are extracted.

- If extracted patterns belong to only one class, then this class is given as answer.
- In other case - the class, which has maximal sum of support by extracted patterns of this class, is given as answer.

$$Q = (?|1,2,1,2)$$

	Pattern set	P intersect. Q	IP(P,Q)	Support	Support set
	Class 1				
P8	(1| 1, 2, –, 1)	(?| 1, 2, –, –)	**0.667**	3	**{R1,R2,R5}**
P9	(1| –, –, 4, –)	(?| –, –, –, –)	0	4	{R1,R4,R5,R6}
P10	(1| –, –, 3, –)	(?| –, –, –, –)	0	2	{R2,R3}
	Class 2				
P7	(2| 3, 1, 2, 2)	(?| –, –, –, 2)	0.250	1	{R9}
P12	(2| –, 1, 1, 2)	(?| –, –, 1, 2)	**0.667**	2	{R7,R8}

"+" advantages

- parameter free algorithm
- very good accuracy for clear datasets

"-" disadvantages

- exponential growth of operations during the process of creating the pattern set
- for big datasets the pattern set is not so compact

Experiments

Lenses dataset

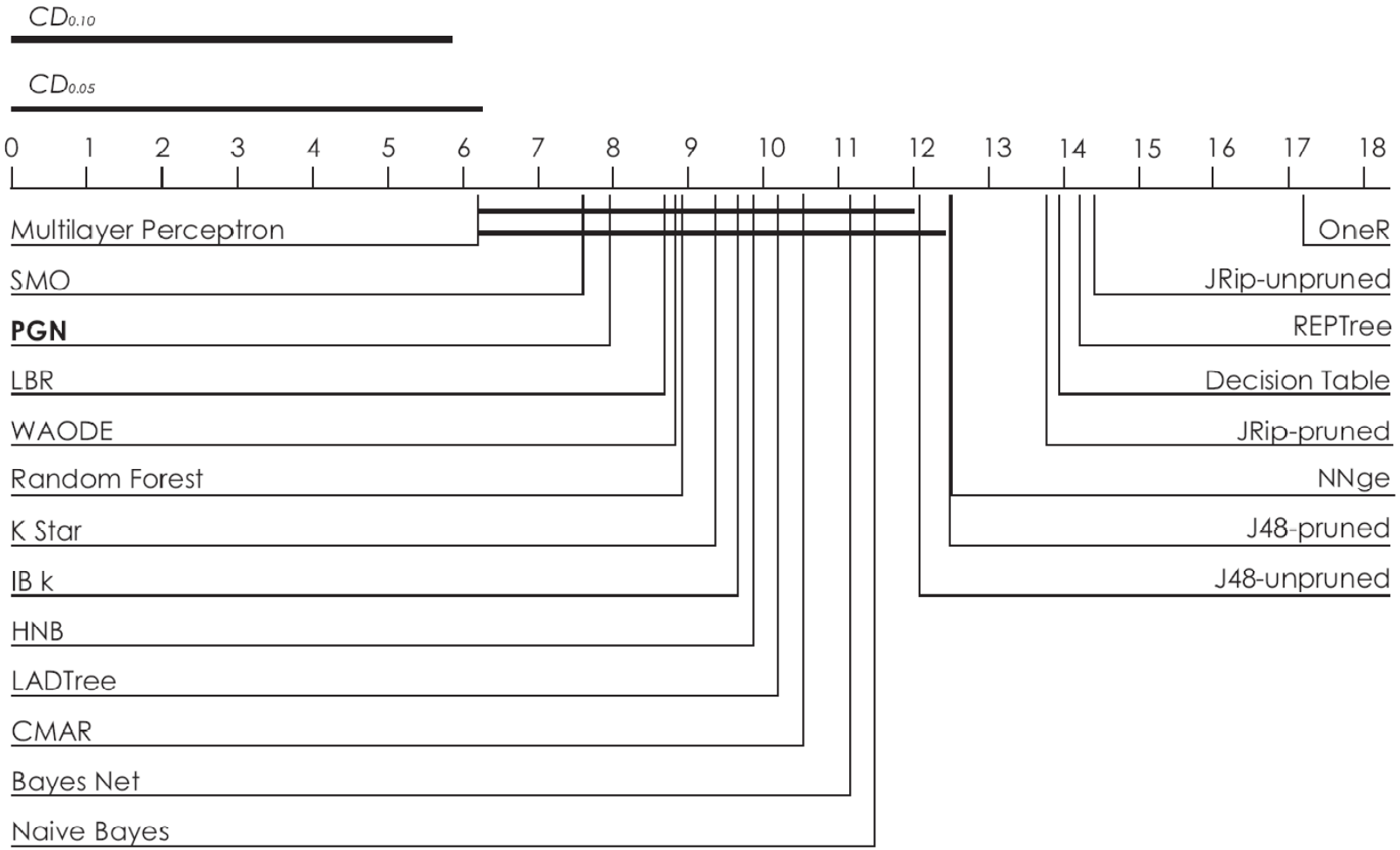
Object	class	age	prescription	astigmatic	tears
1	none	young	myope	no	reduced
2	soft	young	myope	no	normal
3	none	young	myope	yes	reduced
4	hard	young	myope	yes	normal
5	none	young	hypermetrope	no	reduced
6	soft	young	hypermetrope	no	normal
7	none	young	hypermetrope	yes	reduced
8	hard	young	hypermetrope	yes	normal
9	none	pre-presbyopic	myope	no	reduced
10	soft	pre-presbyopic	myope	no	normal
11	none	pre-presbyopic	myope	yes	reduced
12	hard	pre-presbyopic	myope	yes	normal
13	none	pre-presbyopic	hypermetrope	no	reduced
14	soft	pre-presbyopic	hypermetrope	no	normal
15	none	pre-presbyopic	hypermetrope	yes	reduced
16	none	pre-presbyopic	hypermetrope	yes	normal
17	none	presbyopic	myope	no	reduced
18	none	presbyopic	myope	no	normal
19	none	presbyopic	myope	yes	reduced
20	hard	presbyopic	myope	yes	normal
21	none	presbyopic	hypermetrope	no	reduced
22	soft	presbyopic	hypermetrope	no	normal
23	none	presbyopic	hypermetrope	yes	reduced
24	none	presbyopic	hypermetrope	yes	normal

We have achieved 9 rules that are equal to the sufficient set of rules for total description of the Lenses dataset [Cendrowska, 1987].

Rules, produced by PGN					
(2|	_,	_,	_,	2)	Class=none,Tears=reduced
(3|	3,	_,	1,	1)	Class=soft,Age=young,Astigmatic=no,Tears=normal
(1|	3,	_,	2,	1)	Class=hard,Age=young,Astigmatic=yes,Tears=normal
(1|	_,	2,	2,	1)	Class=hard,Prescription=myope,Astigmatic=yes,Tears=normal
(3|	_,	1,	1,	1)	Class=soft,Prescription=hypermetrope,Astigmatic=no,Tears=normal
(3|	1,	_,	1,	1)	Class=soft,Age=pre-presbyopic,Astigmatic=no,Tears=normal
(2|	1,	1,	2,	_)	Class=none,Age=pre-presbyopic,Prescription=hypermetrope,Astigmatic=yes
(2|	2,	2,	1,	_)	Class=none,Age=presbyopic,Prescription=myope,Astigmatic=no
(2|	2,	1,	2,	_)	Class=none,Age=presbyopic,Prescription=hypermetrope,Astigmatic=yes

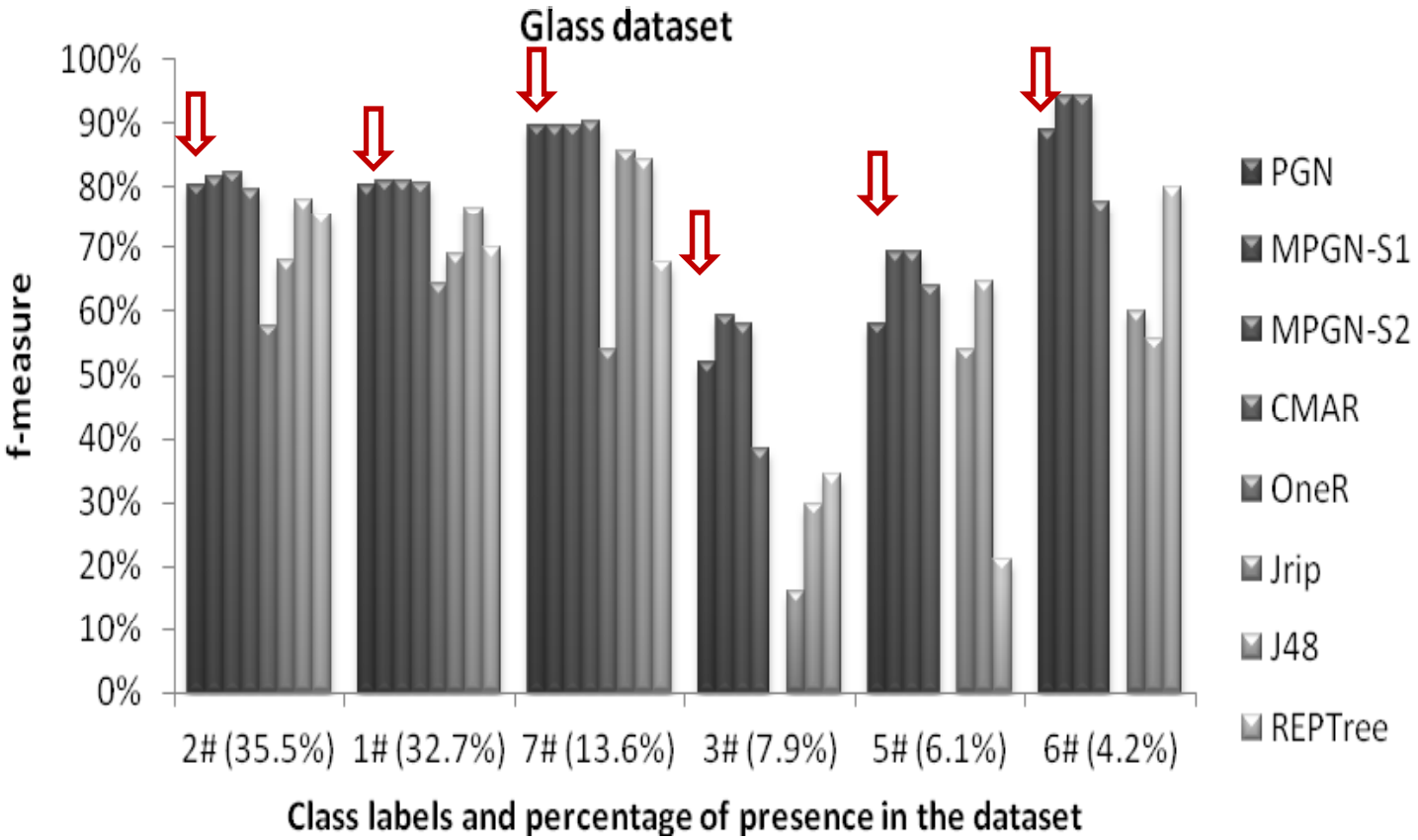
Global accuracy

25 datasets; 21 classifiers



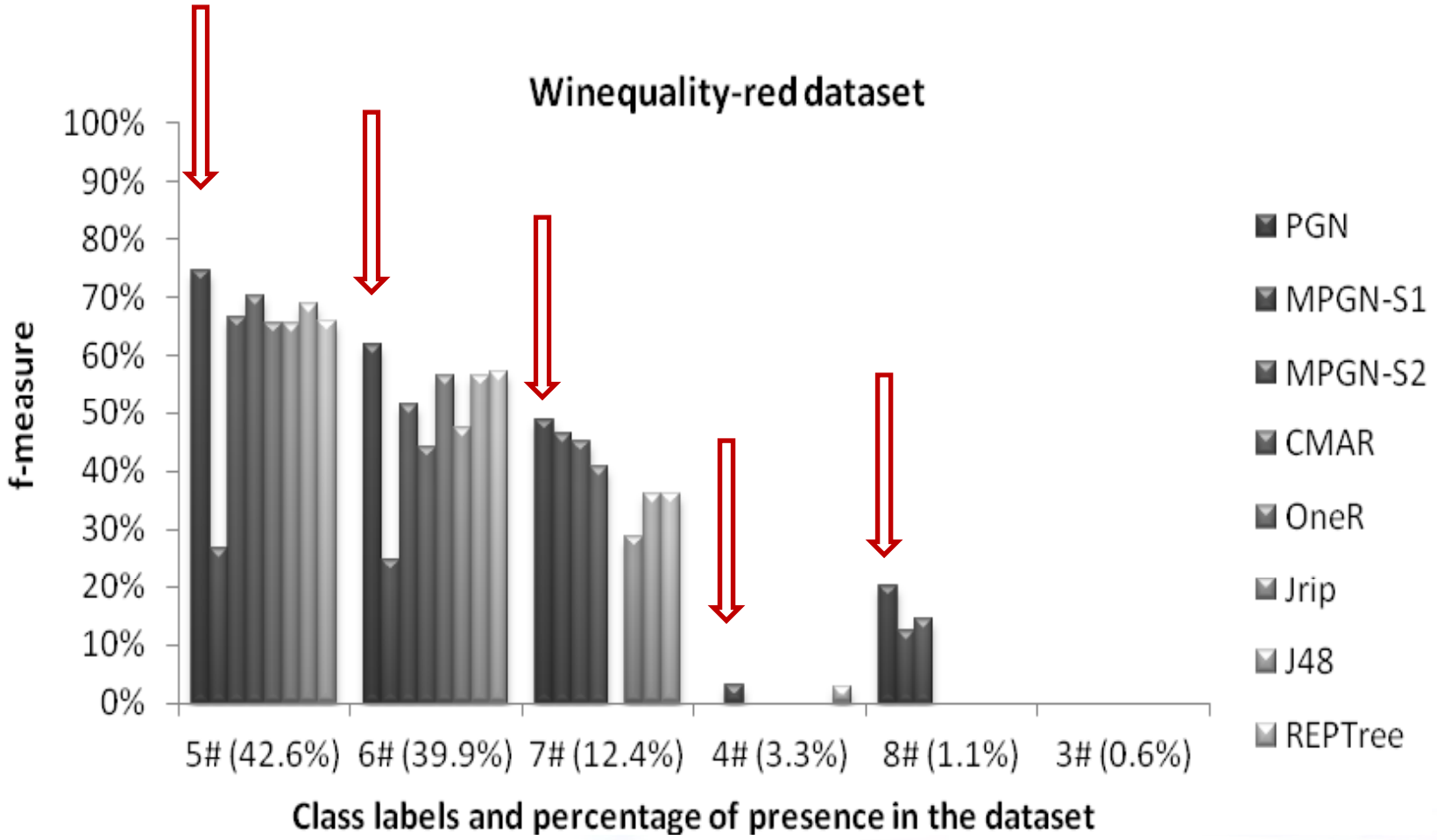
F-measure

Dataset with multiple classes



F-measure

Dataset with multiple classes and non-uniform distribution

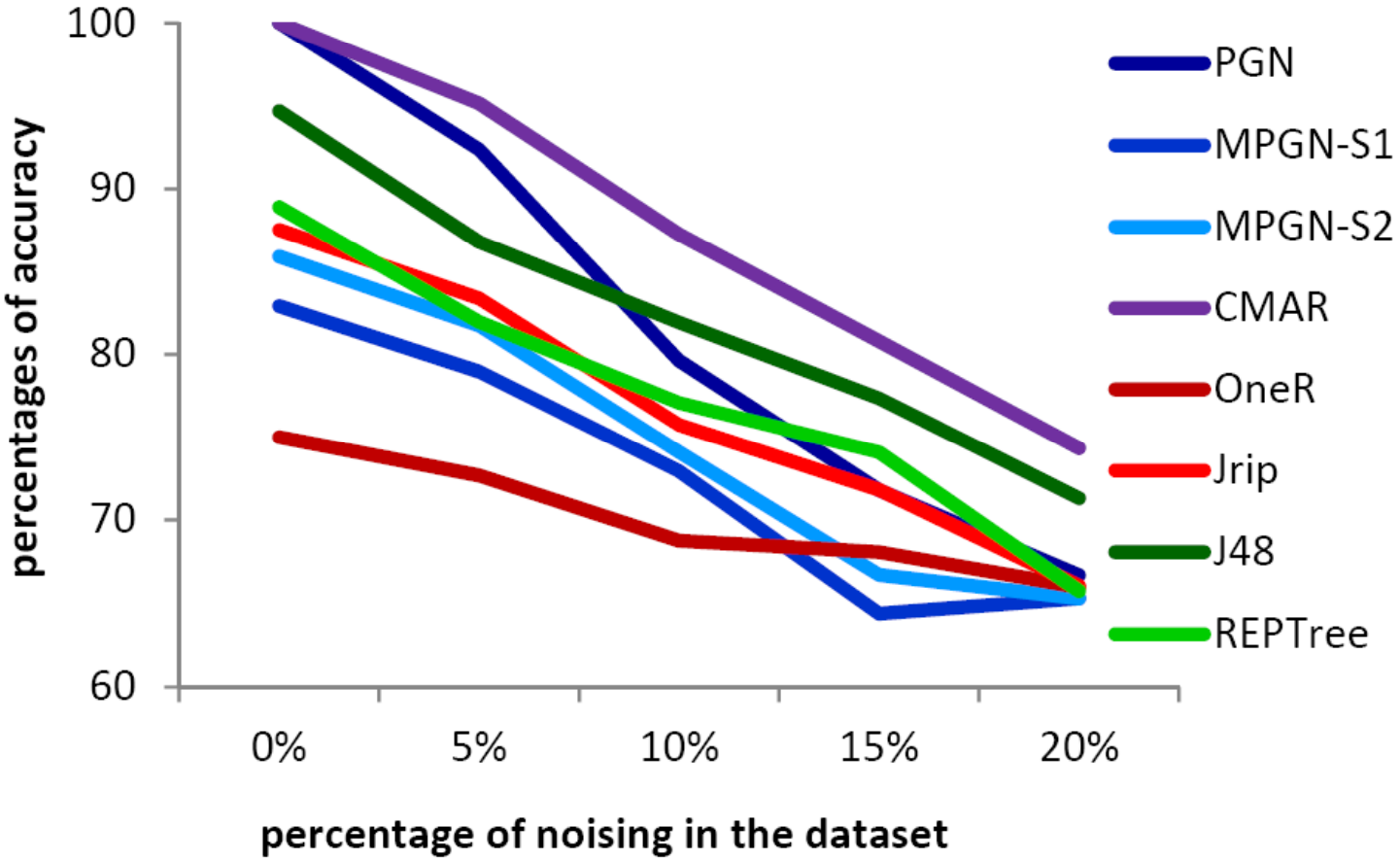


Noise

- The noising of the datasets is made by choosing random instance and attribute and replacing the value with arbitrary chosen possible for this attribute values (without repetitive changing of the same positions). Such replacing is made until a desired percentage of noising is achieved.
- Noising within attributes reflects to noising of class labels because of the appearance of contradictory instances.

Percentage of noising in attributes	Resulting noise between class labels
0%	0.00 %
5%	6.00 %
10%	12.50 %
15%	17.25 %
20%	22.45 %

Noise – example Monks1



Conclusion

PGN:

+ Competitive with classifiers in the group of SVM and Neural Networks
+ Statistically outperforms trees and decision rules
+ Better recognition of multiclass and non-uniform datasets
– Noise sensitive
– Exponential problem

Amendable to parallelization

References

Mitov I., B. Depaire, K. Ivanova, K. Vanhoof, Classifier PGN: Classification with high confidence rules, Serdica J. Computing 6(2), 2013, 143-164.

Vanhoof K., B. Depaire, Structure of Association Rule Classifiers: a Review, Proc. of the Int. Conf. “Intelligent Systems and Knowledge Engineering” (ISKE), Hangzhou, Publ. IEEE, 2010, 9-12.