QCBA: improving rule classifiers learned from quantitative data by recovering information lost by discretisation

Presentation by Tomáš Kliegr

Department of Information and Knowledge Engineering,

VSE v Praze

RuleML Webinar 26.4.2023

Association rule-based classification models

The Apriori algorithm was considered a major breakthrough soon after its publication in 1994

„ ... The apriori algorithm represents one of the major advances in data mining technology.“

Hastie et al. Elements of Statistical Learning, p. 490

The algorithm can process large datasets in short time.

In 1998, the algorithm was extended for supervised classification:

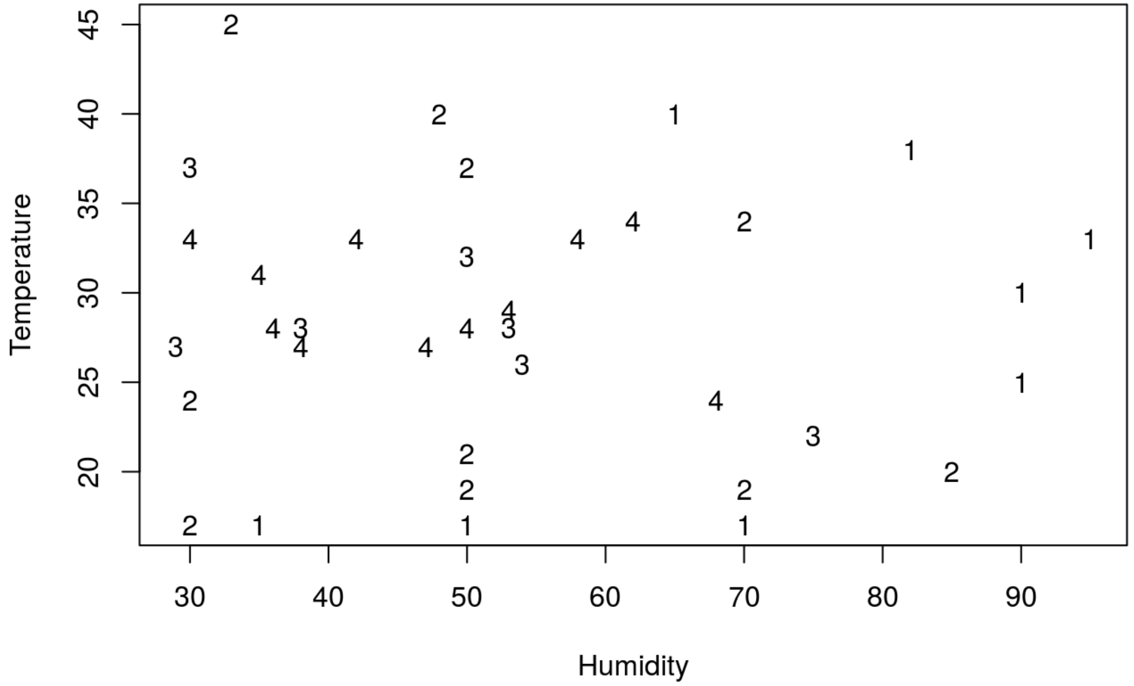
*Ma, Bing Liu Wynne Hsu Yiming, and Bing Liu. "**Integrating classification and association rule mining**." Proceedings of the fourth international conference on knowledge discovery and data mining. 1998.*

Illustration problem

Dataset contains historical data on worker's comfort

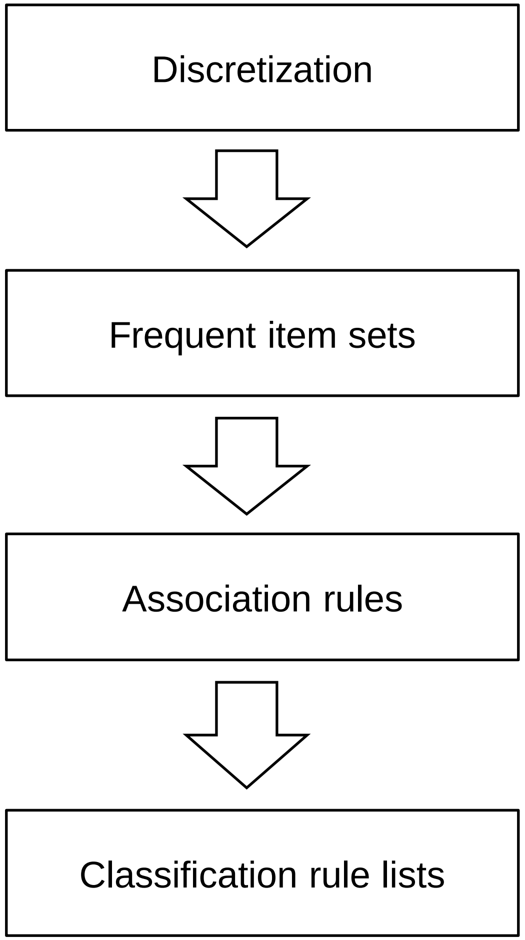
- Two predictors: temperature (Y axis) and room humidity (X axis)
- One target attribute: worker's comfort (1 = worst, 4 = best)

The dataset was designed to allow visualization in 2D



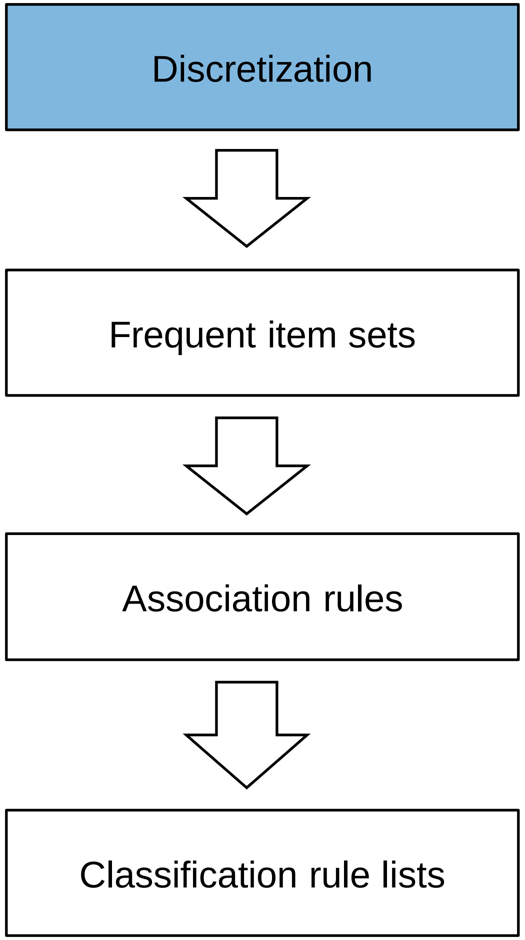
Classification based on Associations

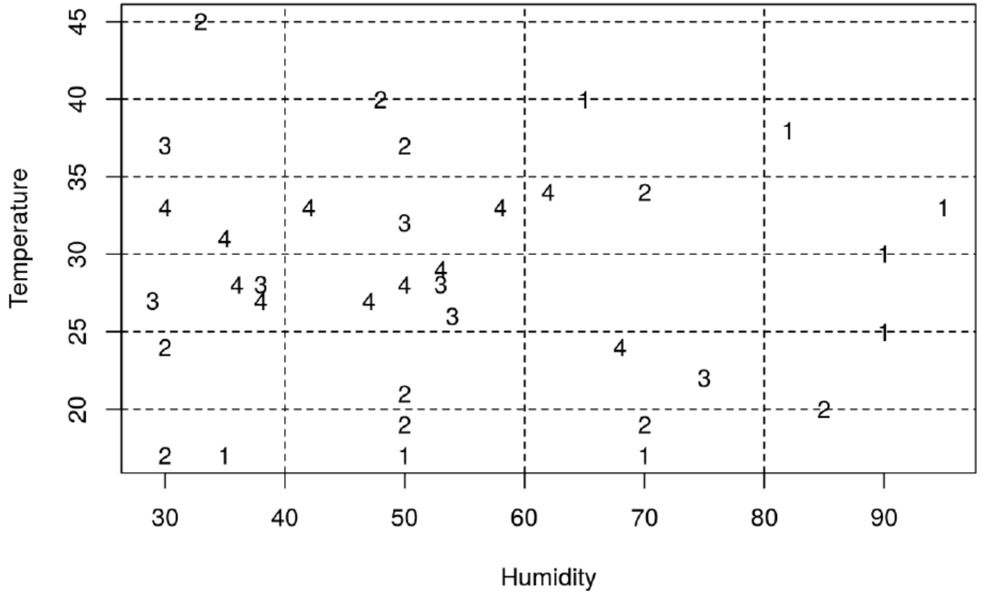
principle of the CBA algorithm (Liu, 1998)



Classification based on Associations (CBA)

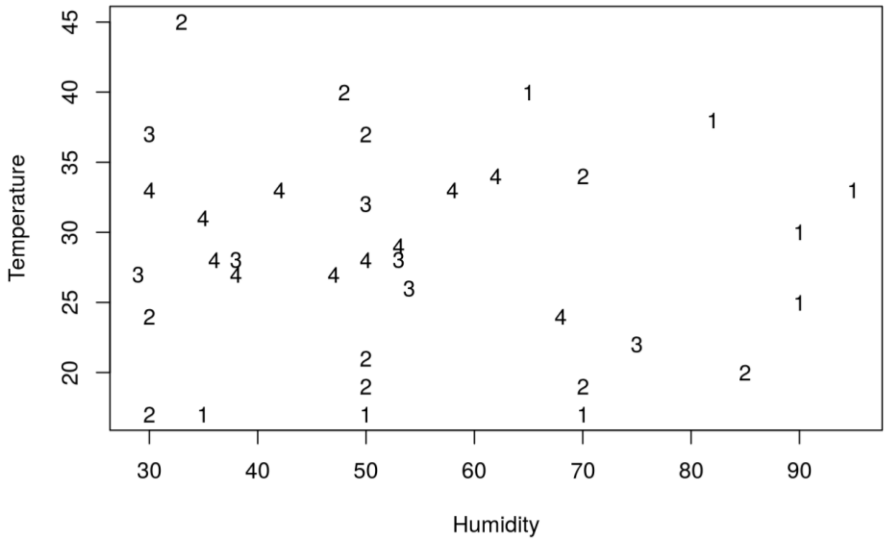
only nominal attributes are on the input

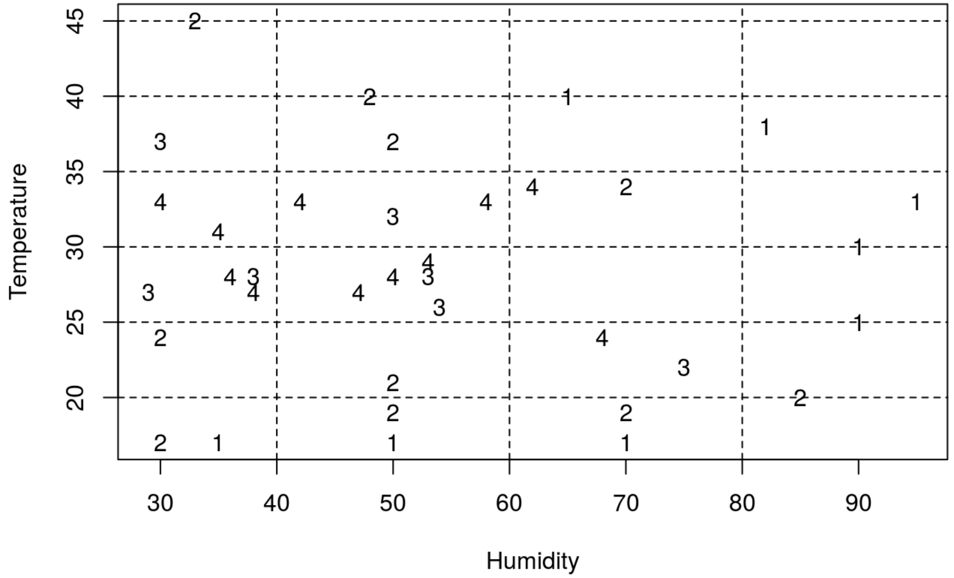




- Algorithms for association rule mining accept only nominal attributes on the input.
- For discretization – conversion of numerical attributes to intervals – one typically uses equidistant method or the entropy-based MDLP algorithm (Fayyad, 93)
- Item is a tuple: `attribute=value`
  `Humidity=(40;60]`

Discretization



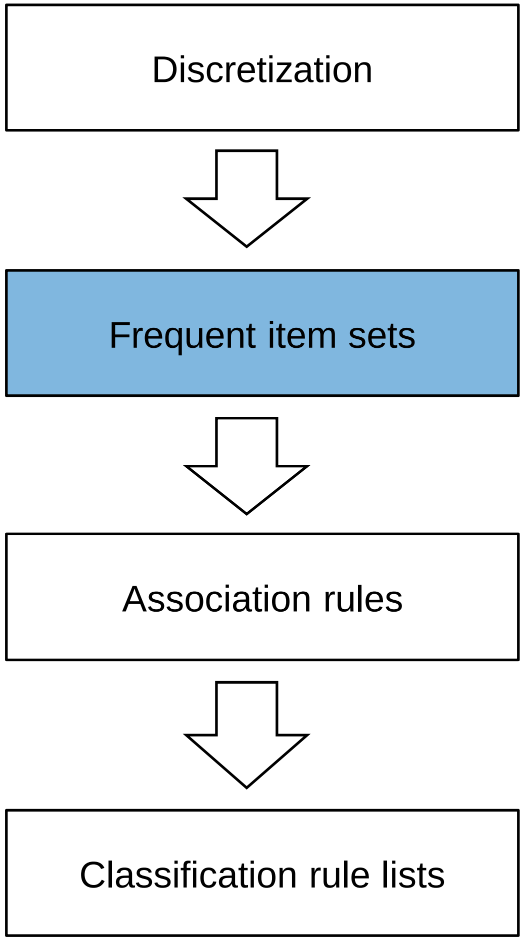


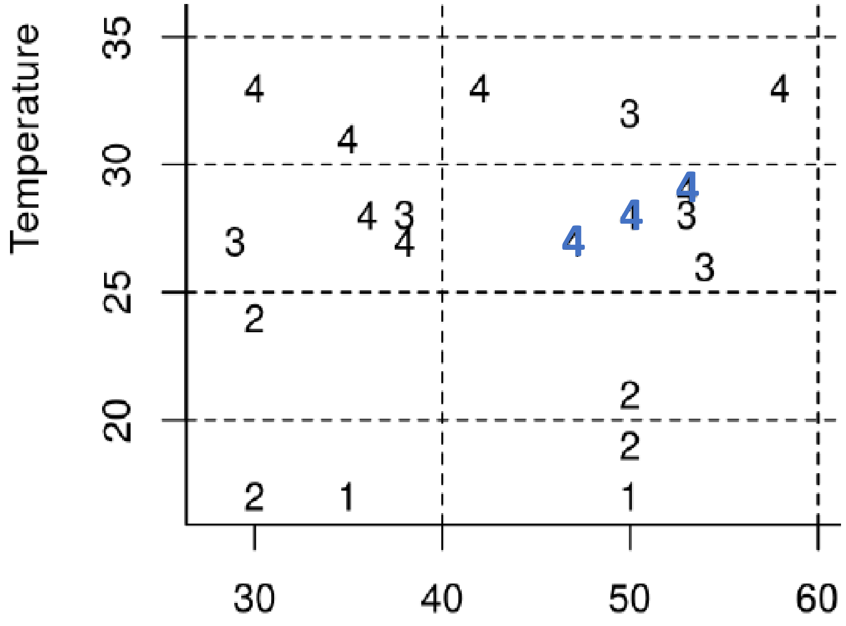
Temperature	Humidity	Class
45	33	2
27	29	3
40	48	2
40	65	1
38	82	1
37	30	3

Temperature	Humidity	Class
(40;45]	(0;40]	2
(25;30]	(0;40]	3
(35;40]	(40;60]	2
(35;40]	(60;80]	1
(35;40]	(80;100]	1
(35;40]	(0;40]	3

Classification based on Associations (CBA)

support of item set





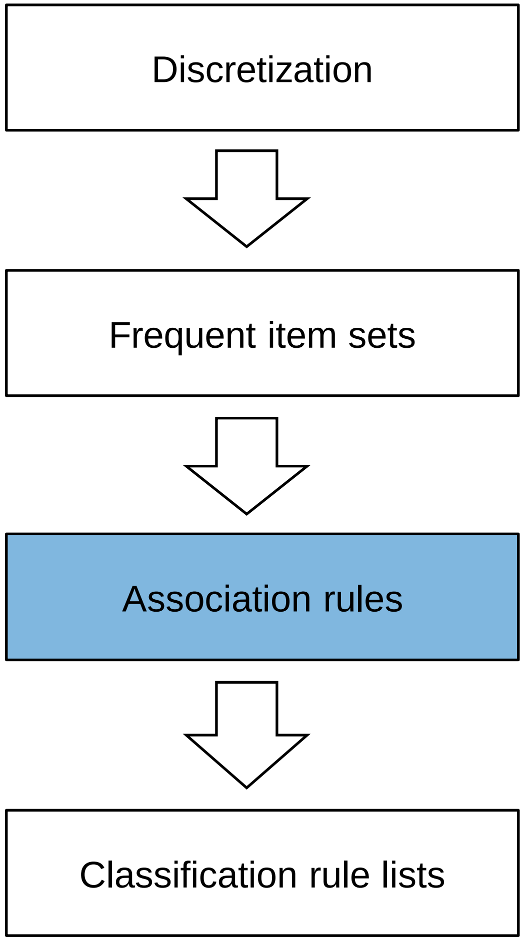
Item set = conjunction of conditions

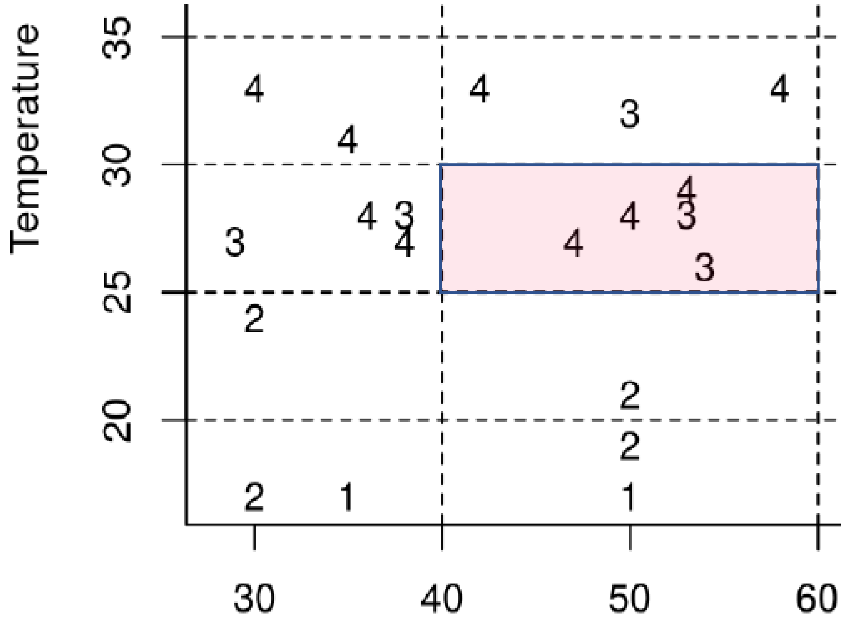
`Temp=(25;30] AND Hum=(40;60] AND Comf=4;`
`support = 3`

Minimum support: algorithm finds all combinations of items, which are *frequent* - they appear in at least user-set minimum number of input rows.

Classification based on Associations (CBA)

confidence of association rule





`Temp=(25;30] AND Hum=(40;60] => Comf=4`

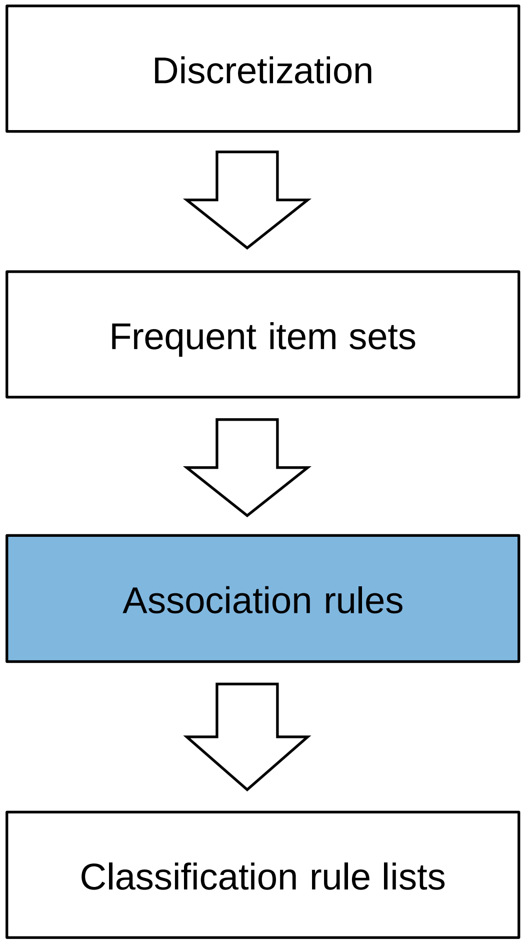
`Support = 3;` **`Confidence = 0.6 = 3/5`**

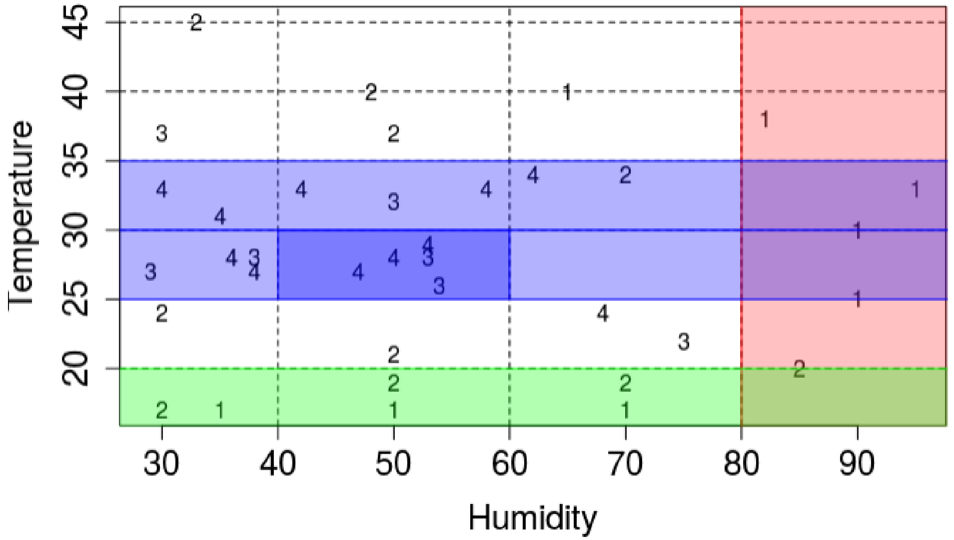
Discovered rules must comply to user-set threshold for **minimum confidence:**

$$\mathrm{Conf}(X \rightarrow Y) = \frac{\text{Number of rows matching X i Y}}{\text{Number of rows matching X}}$$

Classification based on Associations (CBA)

rules are created from frequent item sets





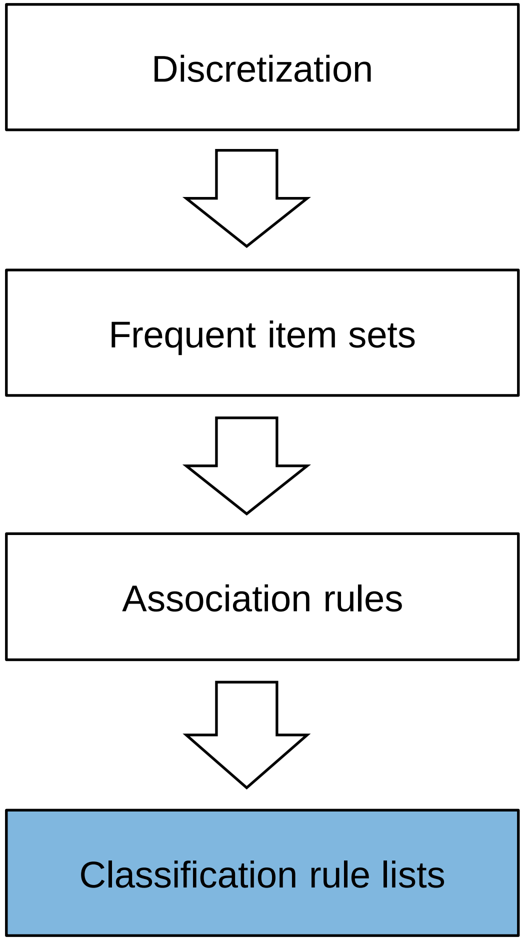
Discovered rules, colours – predicted comfort
minimum confidence = 0.5

1= red, 2 = green, 3 = unassigned, 4 = blue

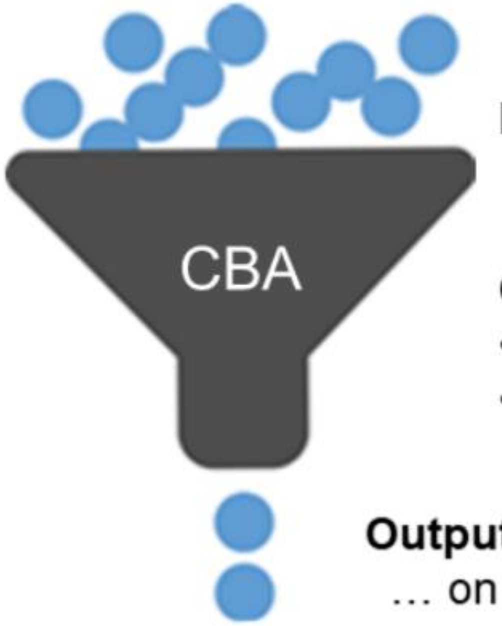
```
{Humidity=(80;100]}                      => {Comfort=1}
{Temperature=(30;35]}                    => {Comfort=4}
{Temperature=(25;30],Humidity=(40;60]}=> {Comfort=4}
{Temperature=(15;20]}                    => {Comfort=2}
{Temperature=(25;30]}                    => {Comfort=4}
```

Classification based on Associations (CBA)

the core of CBA is the effective choice of rules



Input: Discovered association rules
… recommended is 70.000 rules



Principle:
Rules selected using heuristics

Objectives
- Improve accuracy
- Reduce number of rules

Output: Classification rule list
… on average about 70 rules

Choosing rules in CBA

1. Rules are **sorted** according to confidence (higher is better), in case of a tie by support (higher is better), and in case of a tie by length (shorter is better).

2. **Data coverage pruning**: the algorithm iterates through the rules in the sort order removing any rule which does not correctly cover any instances. If the rule correctly covers at least one instance, it is retained and the instances removed.

4. **Default rule pruning**: the algorithm iterates through the rules in the sort order, and cuts off the list once keeping the current rule would result in worse accuracy of the model than if a default rule was inserted at the place of the current rule and the rules below it were removed.

5. **Default rule is inserted** at the bottom of the list. A default rule is a rule with empty antecedent and consequent predicting a majority class left in the training data once the previous rules in the classifier were applied.

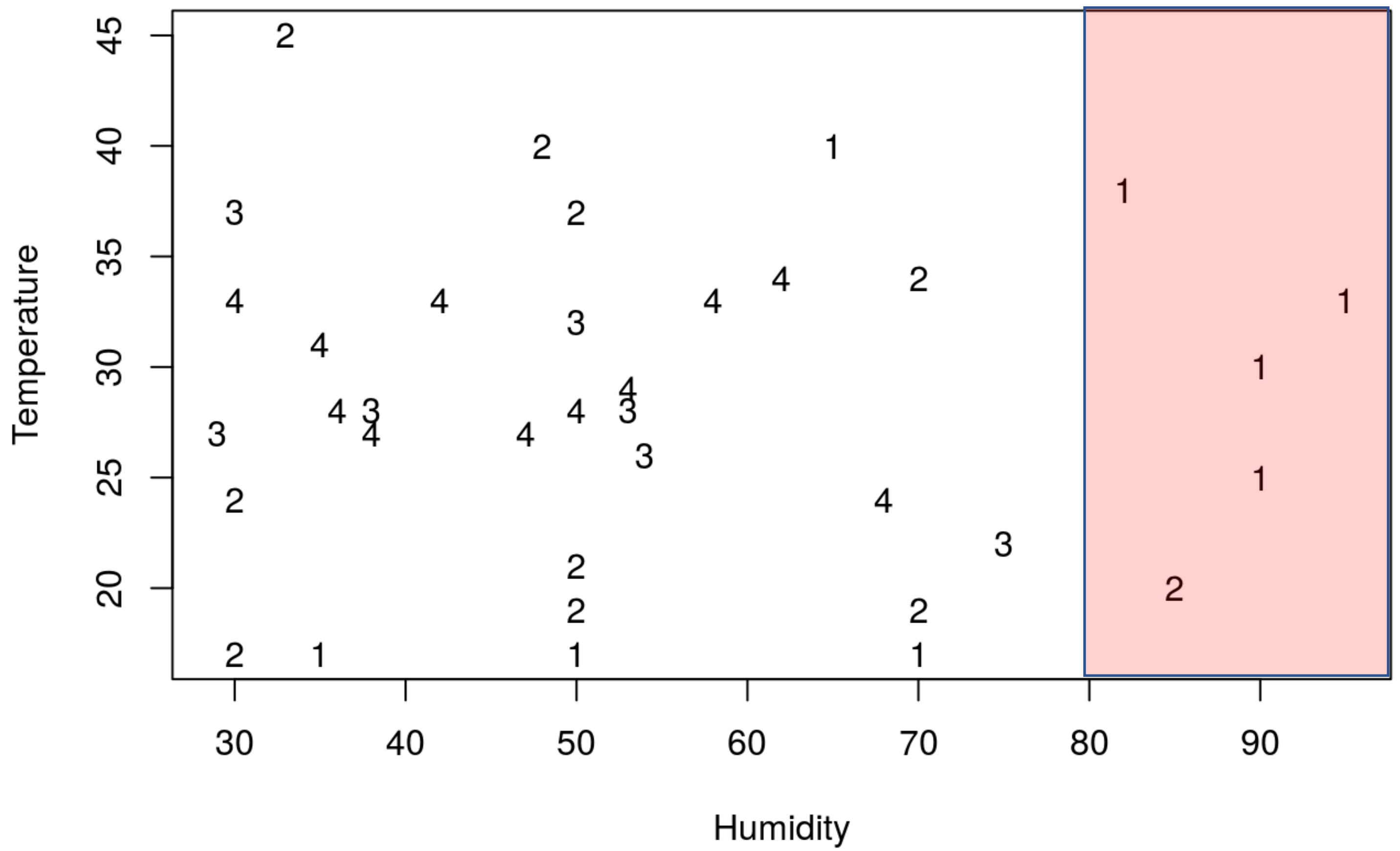
Rules are reordered

```
#       lhs                                    rhs         support    confidence
## [1] {Humidity=(80;100]}                  => {Class=1} 0.11111111 0.8000000
## [2] {Temperature=(15;20]}                => {Class=2} 0.11111111 0.5714286
## [3] {Temperature=(30;35]}                => {Class=4} 0.13888889 0.6250000
## [4] {Temperature=(25;30]}                => {Class=4} 0.13888889 0.5000000
## [5] {Temperature=(25;30],Humidity=(40;60]} => {Class=4} 0.08333333 0.6000000
```

```
##     lhs                                        rhs          sup  conf  len
## [1] {Humidity=(80;100]}                     => {Comfort=1} 0.11  0.80  1
## [2] {Temperature=(30;35]}                   => {Comfort=4} 0.14  0.64  1
## [3] {Temperature=(25;30],Humidity=(40;60]} => {Comfort=4} 0.08  0.60  2
## [4] {Temperature=(15;20]}                   => {Comfort=2} 0.11  0.57  1
## [5] {Temperature=(25;30]}                   => {Comfort=4} 0.14  0.50  1
```

Data coverage pruning

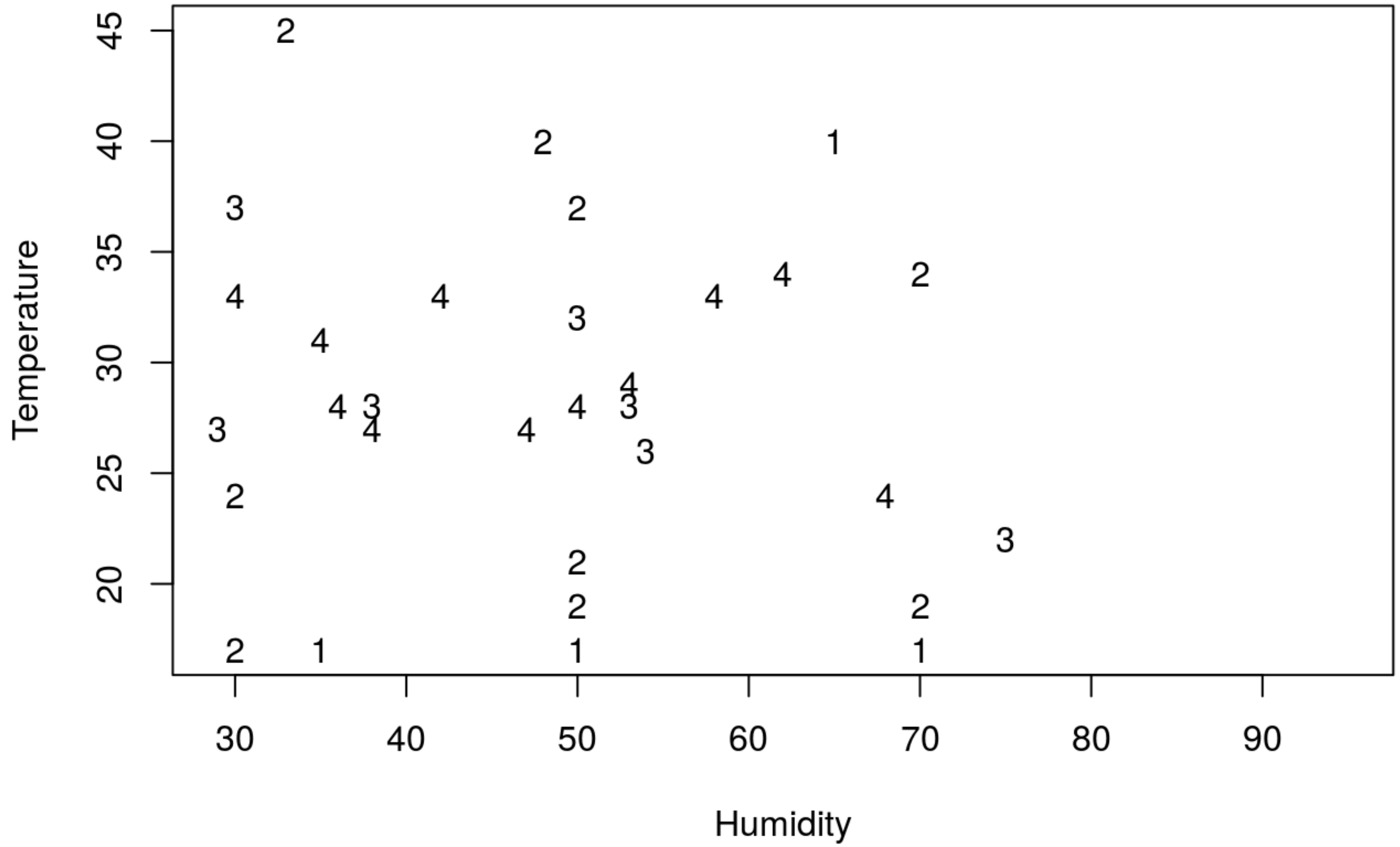
{Humidity=(80;100]} => {Comfort=1}



Rule covers at least one instance correctly – is not removed

Data coverage pruning

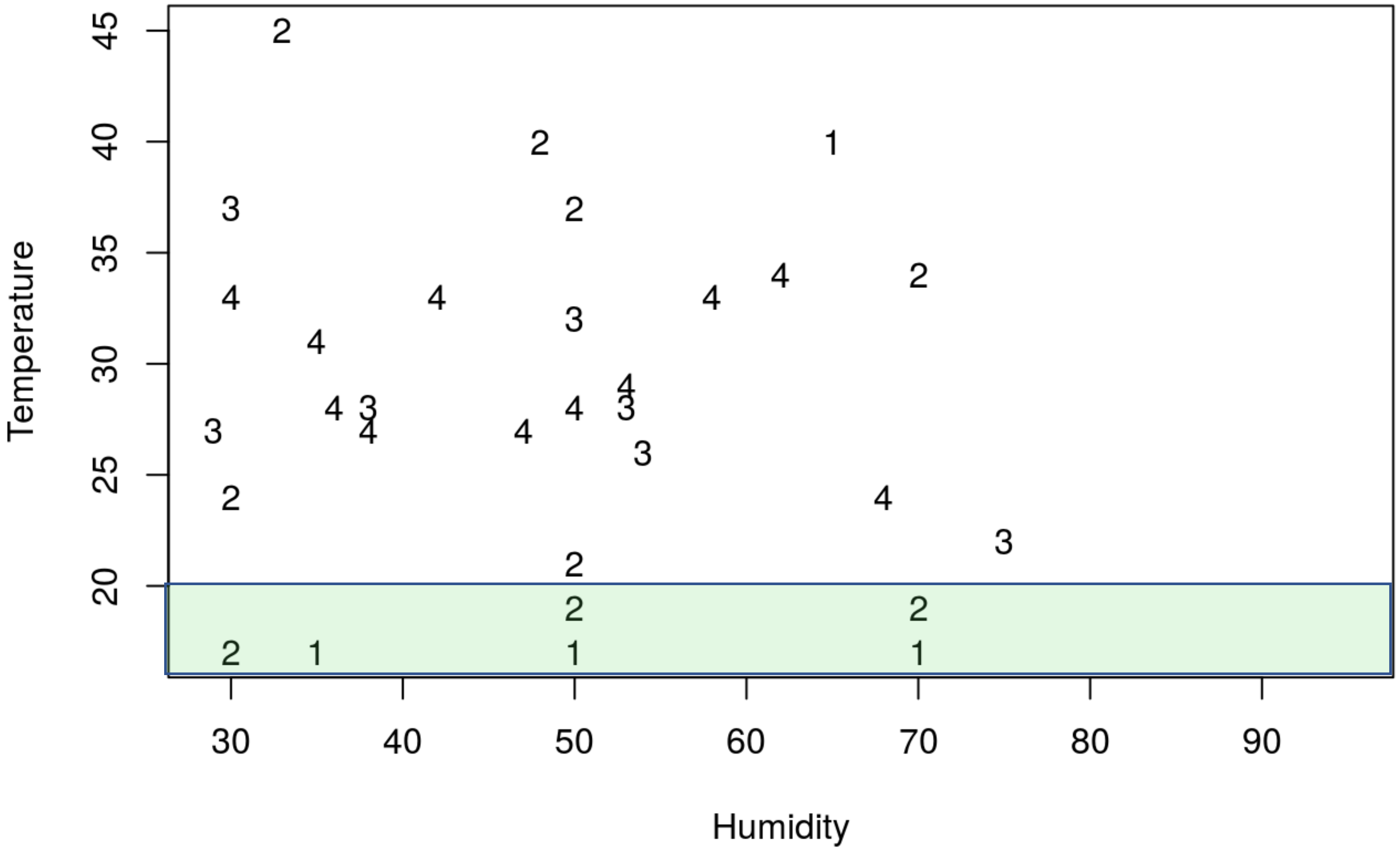
`{Humidity=(80;100]} => {Comfort=1}`



Instances covered by the rule are removed

Data coverage pruning

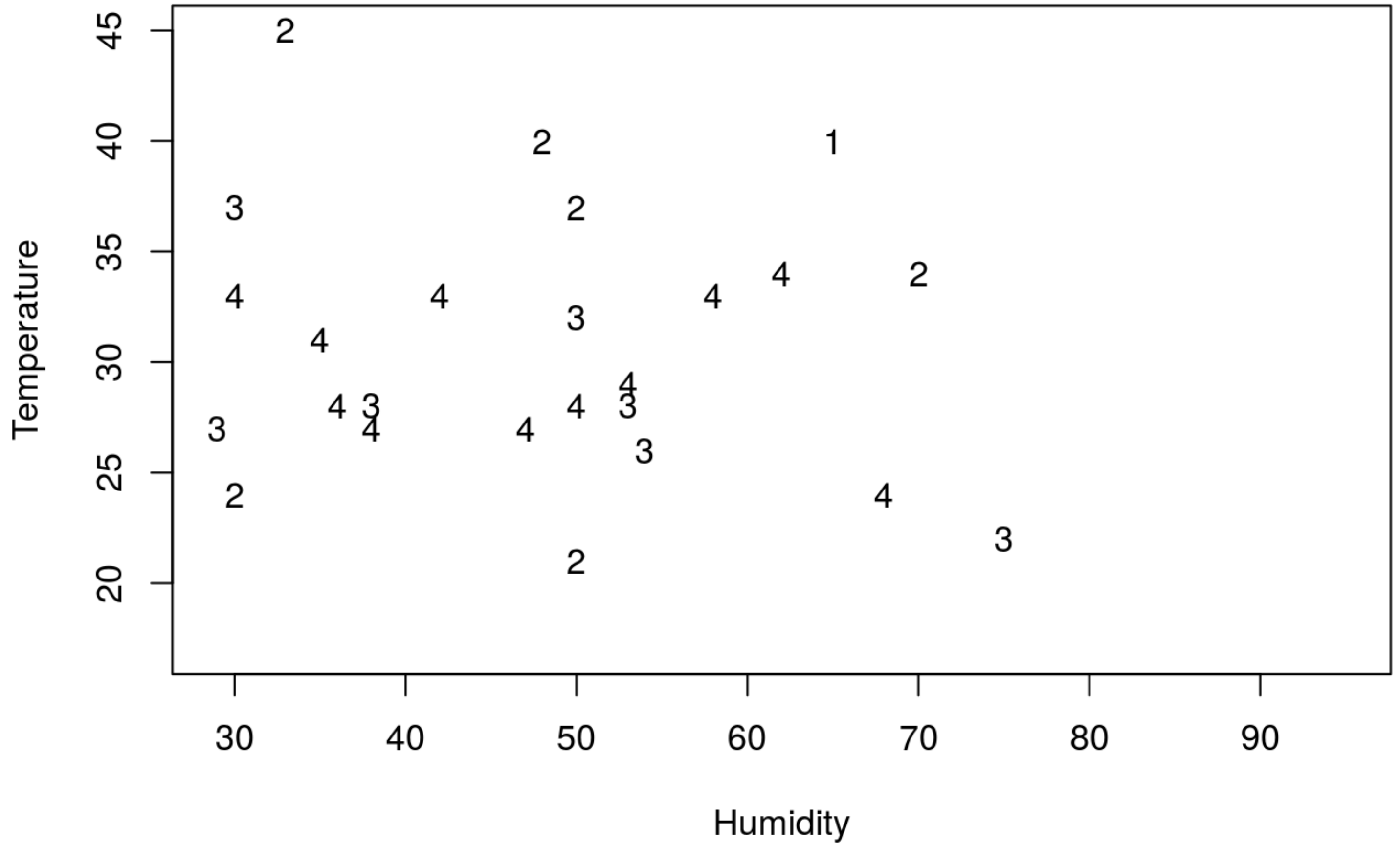
`{Temperature=(15;20]} => {Class=2}`



Rule covers at least one instance correctly – is not removed

Data coverage pruning

`{Temperature=(15;20]} => {Class=2}`



Instances covered by the rule are removed

Data coverage – pruning overview

```
##      lhs                                          rhs           sup  conf  len
## [1]  {Humidity=(80;100]}                       => {Comfort=1} 0.11  0.80  1
## [2]  {Temperature=(30;35]}                     => {Comfort=4} 0.14  0.64  1
## [3]  {Temperature=(25;30],Humidity=(40;60]}    => {Comfort=4} 0.08  0.60  2
## [4]  {Temperature=(15;20]}                     => {Comfort=2} 0.11  0.57  1
## [5]  {Temperature=(25;30]}                     => {Comfort=4} 0.14  0.50  1
```

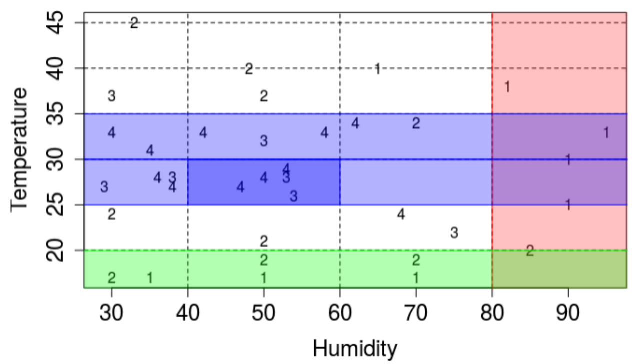
Already processed - retained

Processed analogically, also retained

No rule is pruned in this toy example, the final rule list is the same as input rule list

```
##      lhs                                          rhs           sup  conf  len
## [1]  {Humidity=(80;100]}                       => {Comfort=1} 0.11  0.80  1
## [2]  {Temperature=(30;35]}                     => {Comfort=4} 0.14  0.64  1
## [3]  {Temperature=(25;30],Humidity=(40;60]}    => {Comfort=4} 0.08  0.60  2
## [4]  {Temperature=(15;20]}                     => {Comfort=2} 0.11  0.57  1
## [5]  {Temperature=(25;30]}                     => {Comfort=4} 0.14  0.50  1
```

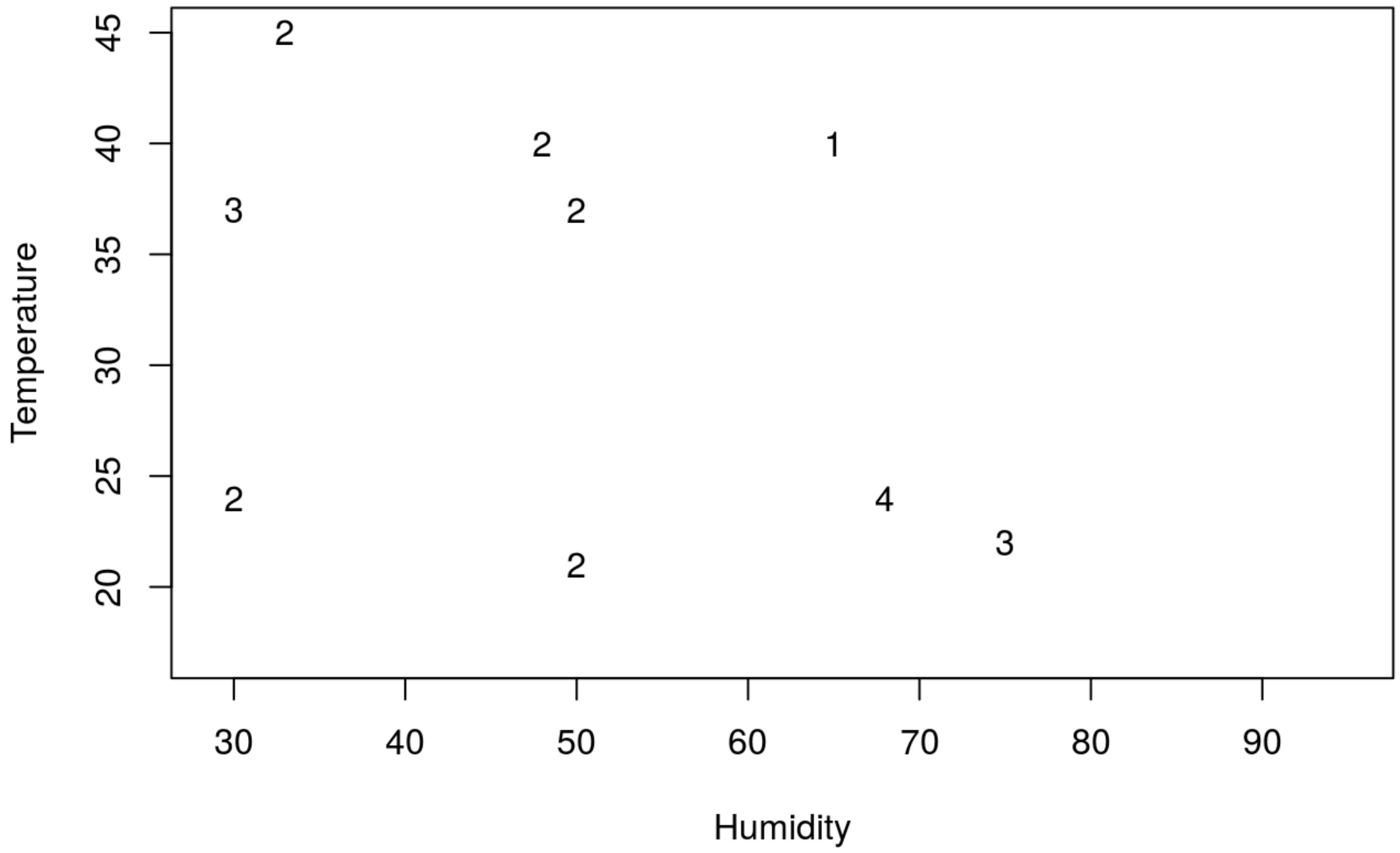
Default rule is computed



The most common class in the remaining instances is 2.

The new default rule is thus:

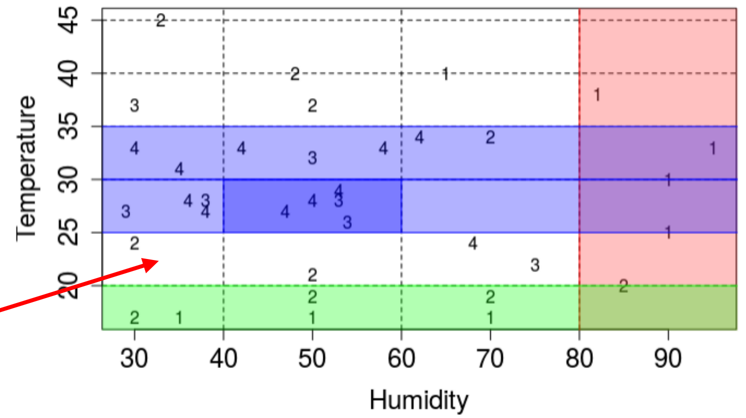
`{} => {Comfort=2}`



This step is here simplified - CBA would try to simplify the rule list by replacing the end of the list by the default rule if that would not increase the error.

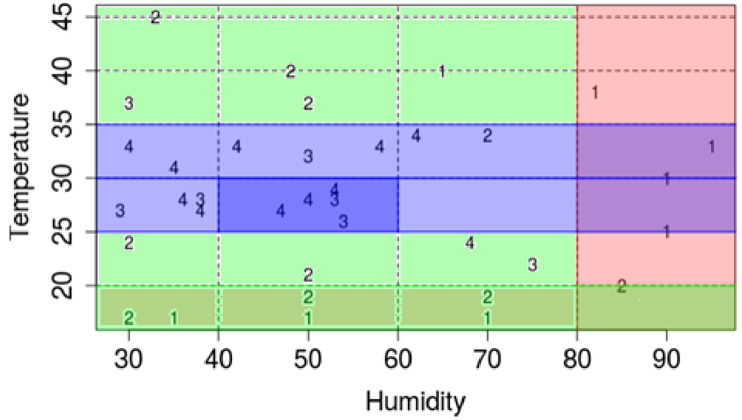
Default rule is added

```
##     lhs                                         rhs             sup  conf  len
## [1] {Humidity=(80;100]}                      => {Comfort=1} 0.11  0.80  1
## [2] {Temperature=(30;35]}                    => {Comfort=4} 0.14  0.64  1
## [3] {Temperature=(25;30],Humidity=(40;60]} => {Comfort=4} 0.08  0.60  2
## [4] {Temperature=(15;20]}                    => {Comfort=2} 0.11  0.57  1
## [5] {Temperature=(25;30]}                    => {Comfort=4} 0.14  0.50  1
```



Instance located in any of the white regions would not be covered by the rule

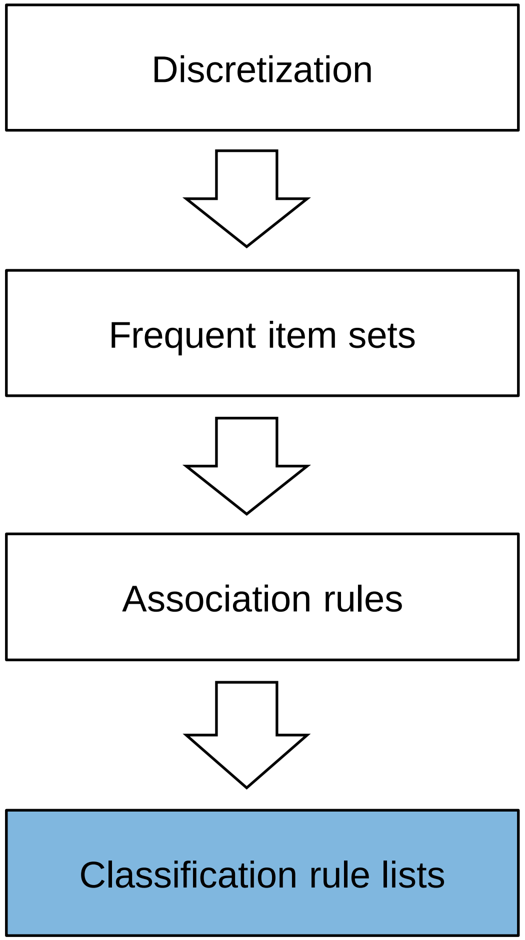
```
##     lhs                                         rhs             sup  conf  len
## [1] {Humidity=(80;100]}                      => {Comfort=1} 0.11  0.80  1
## [2] {Temperature=(30;35]}                    => {Comfort=4} 0.14  0.64  1
## [3] {Temperature=(25;30],Humidity=(40;60]} => {Comfort=4} 0.08  0.60  2
## [4] {Temperature=(15;20]}                    => {Comfort=2} 0.11  0.57  1
## [5] {Temperature=(25;30]}                    => {Comfort=4} 0.14  0.50  1
## [6] {}                                       => {Comfort=2} 0.28  0.28  x
```

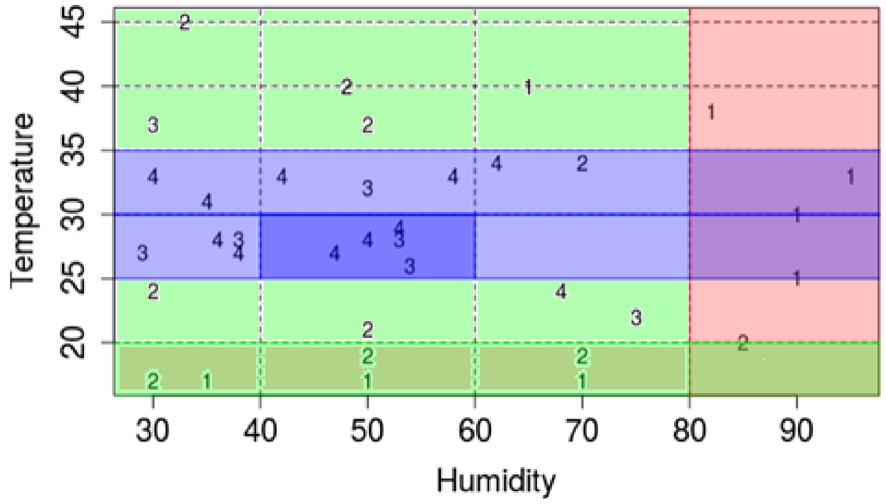


A default „fallback“ rule has been added as the last rule.
This rule covers any instance as it has the list of conditions in the antecedent empty.

Classification based on Associations (CBA)

rule list is used to create the classifier

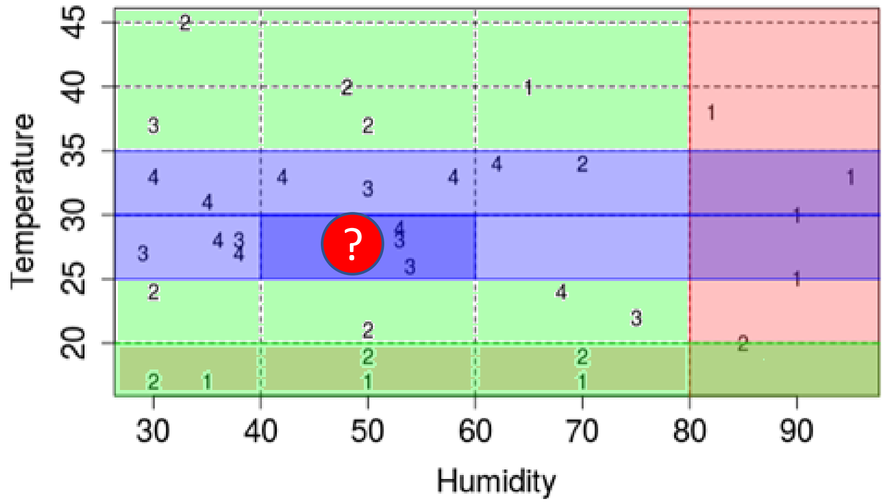




The last rule in the classifier is called default rule (light green), it ensures that all conceivable instances are covered by the classifier.

Classification based on Associations (CBA)

use for prediction



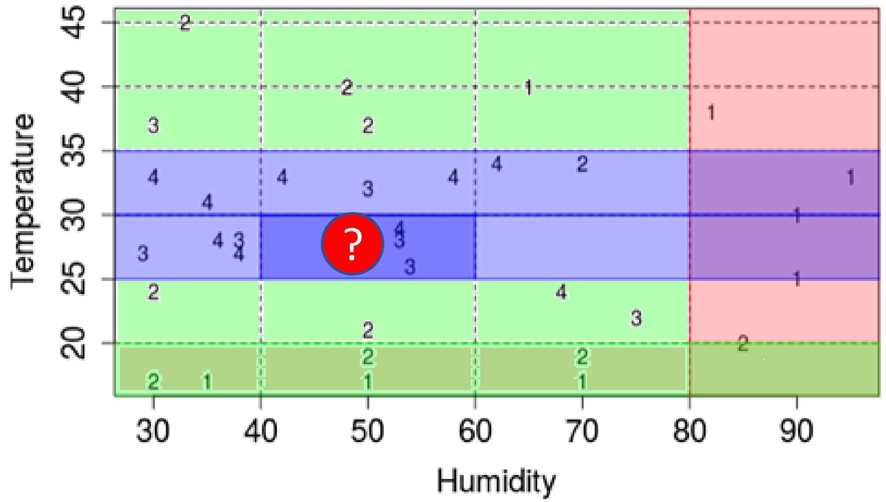
- The first rule in the order of confidence, support and length (more general rules are preferred)

Temperature	Humidity	Comfort
27	48	?

```
##     lhs                                     rhs          sup  conf  len
## [1] {Humidity=(80;100]}                  => {Comfort=1} 0.11  0.80  1
## [2] {Temperature=(30;35]}                => {Comfort=4} 0.14  0.64  1
## [3] {Temperature=(25;30],Humidity=(40;60]} => {Comfort=4} 0.08  0.60  2
## [4] {Temperature=(15;20]}                => {Comfort=2} 0.11  0.57  1
## [5] {Temperature=(25;30]}                => {Comfort=4} 0.14  0.50  1
## [6] {}                                   => {Comfort=2} 0.28  0.28  x
```

Classification based on Associations (CBA)

use for prediction

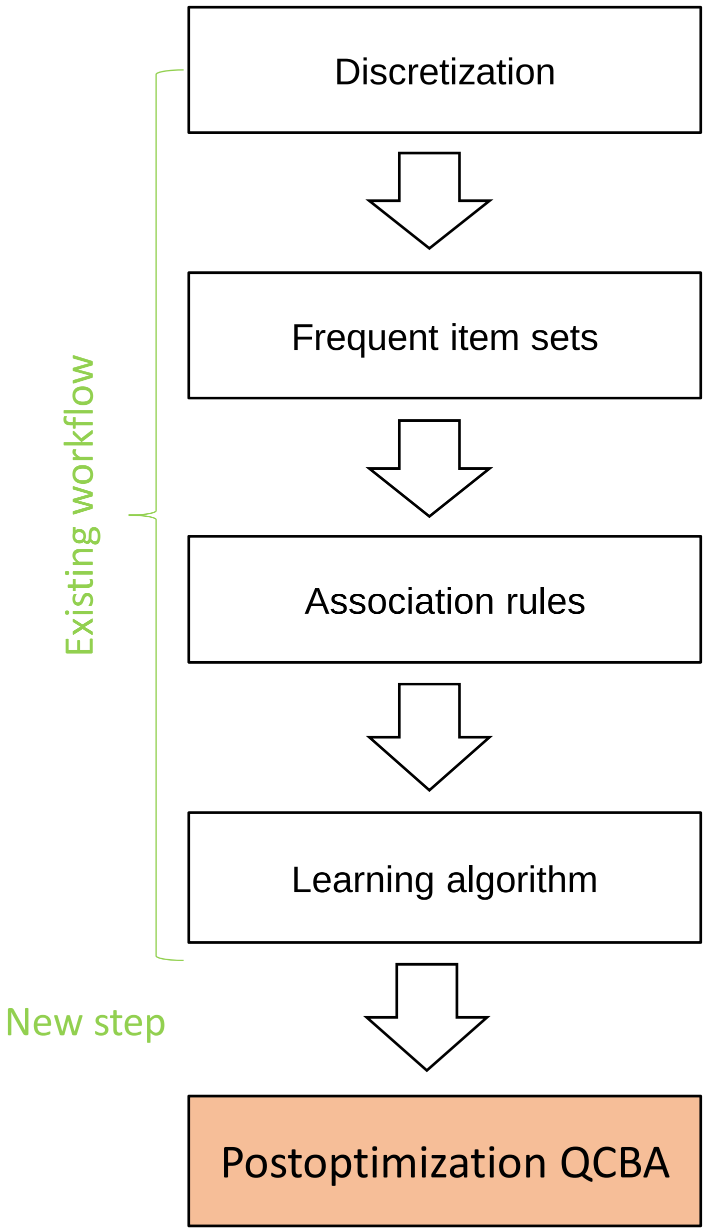


- The first rule in the order of confidence, support and length (more general rules are preferred)

Temperature	Humidity	Comfort
27	48	4

```
##      lhs                                           rhs          sup  conf  len
## [1] {Humidity=(80;100]}                        => {Comfort=1} 0.11  0.80  1
## [2] {Temperature=(30;35]}                      => {Comfort=4} 0.14  0.64  1
## [3] {Temperature=(25;30],Humidity=(40;60]}     => {Comfort=4} 0.08  0.60  2
## [4] {Temperature=(15;20]}                      => {Comfort=2} 0.11  0.57  1
## [5] {Temperature=(25;30]}                      => {Comfort=4} 0.14  0.50  1
## [6] {}                                         => {Comfort=2} 0.28  0.28  x
```

QCBA



Kliegr, T., & Izquierdo, E. (2023). QCBA: improving rule classifiers learned from quantitative data by recovering information lost by discretisation. Applied Intelligence, 1-31.

qcba().

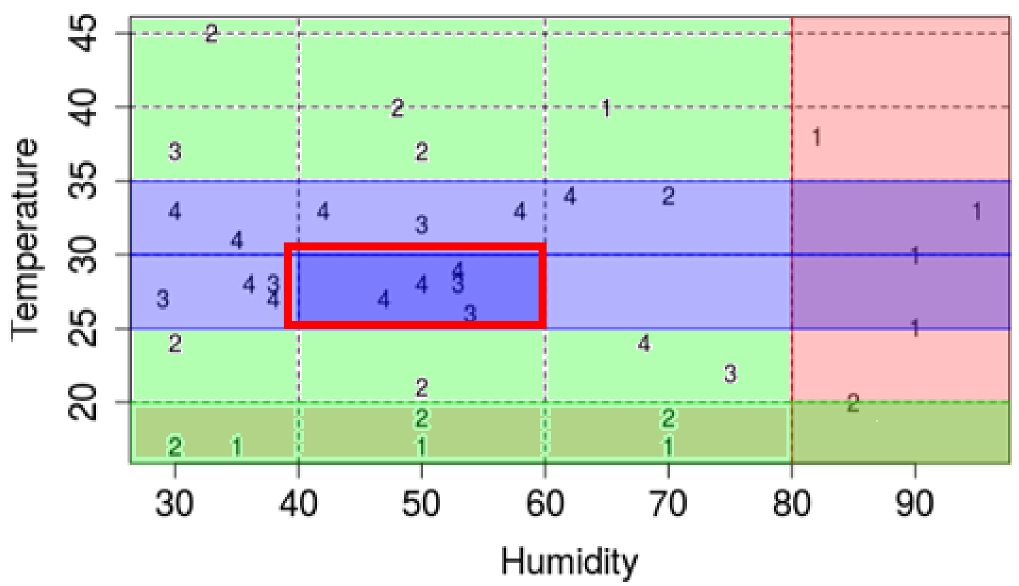
Input: *rules* input rule set (order not important)
Output: *rules* ordered rule list

```
rules ← remove rules with empty antecedent from rules.
for rule ∈ rules do ▷ Can be parallelized
  rule ← refit(rule)
  rule ← pruneLiterals(rule)
  rule ← trim(rule)
  rule ← extendRule(rule)
end for
rules ← postprune(rules)
rules ← drop(rules) ▷ Two versions: instance-based or range-based version
return rules
```

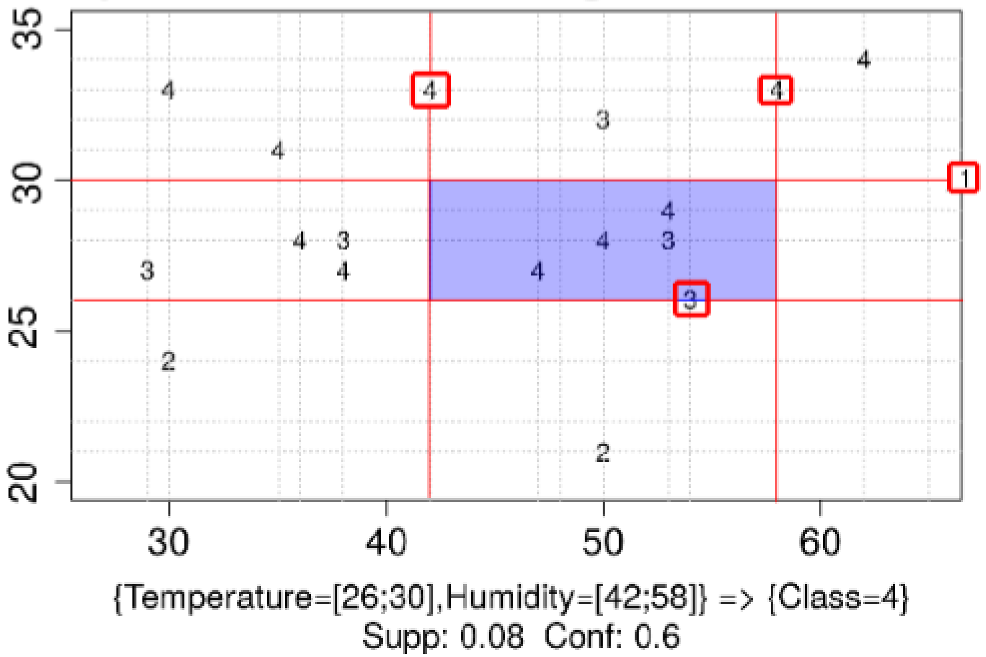
QCBA: Refit step

```
##      lhs                                           rhs
## [1] {Humidity=(80;100]}                         => {Comfort=1}
## [2] {Temperature=(30;35]}                       => {Comfort=4}
## [3] {Temperature=(25;30],Humidity=(40;60]}      => {Comfort=4}
## [4] {Temperature=(15;20]}                       => {Comfort=2}
## [5] {Temperature=(25;30]}                       => {Comfort=4}
## [6] {}                                          => {Comfort=2}
```

Rule in the original grid



Rule is refit to the finer grid



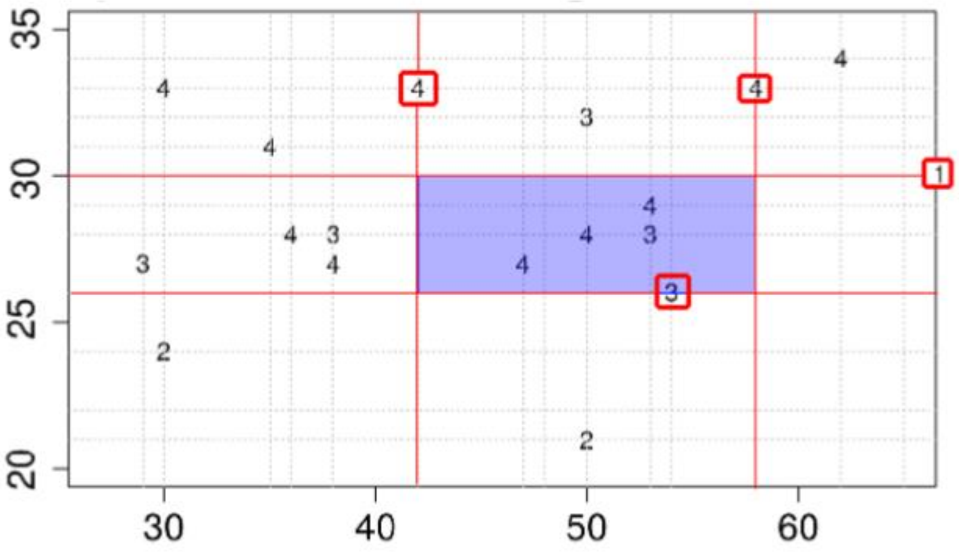
The red boxes in the figure on the right mark the instances that are used as “anchors” for the refit of Rule #3. For the literal “Temperature=(25,30]”, the upper boundary corresponds to an existing instance; therefore, there is no change. Since the lower boundary is exclusive, it is adjusted to the nearest value of a real instance, which is 26. Likewise, the boundaries of Humidity in Rule #3 are adjusted to values of the nearest instances within the original boundaries.

QCBA: Literal pruning

- This is simple step, where the algorithm tries to remove redundant conditions – removal is performed if the confidence does not decrease
- This is not necessary (and ineffective) as a postprocessing for some rule learning algorithms

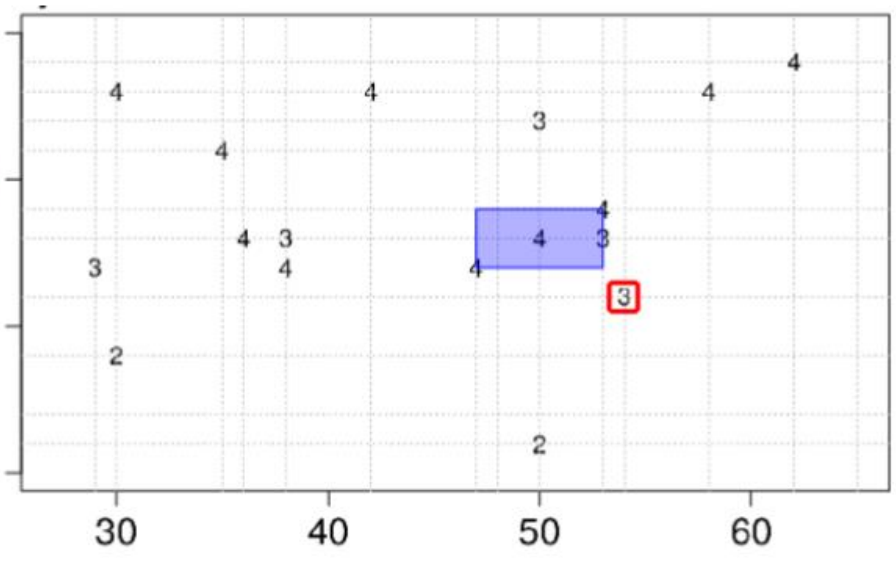
QCBA: Trimming

Rule is refit to the finer grid



{Temperature=[26;30],Humidity=[42;58]} => {Class=4}
Supp: 0.08 Conf: 0.6

Rule is trimmed



{Temperature=[27;29],Humidity=[47;53]} => {Class=4}
Supp: 0.08 Conf: 0.75

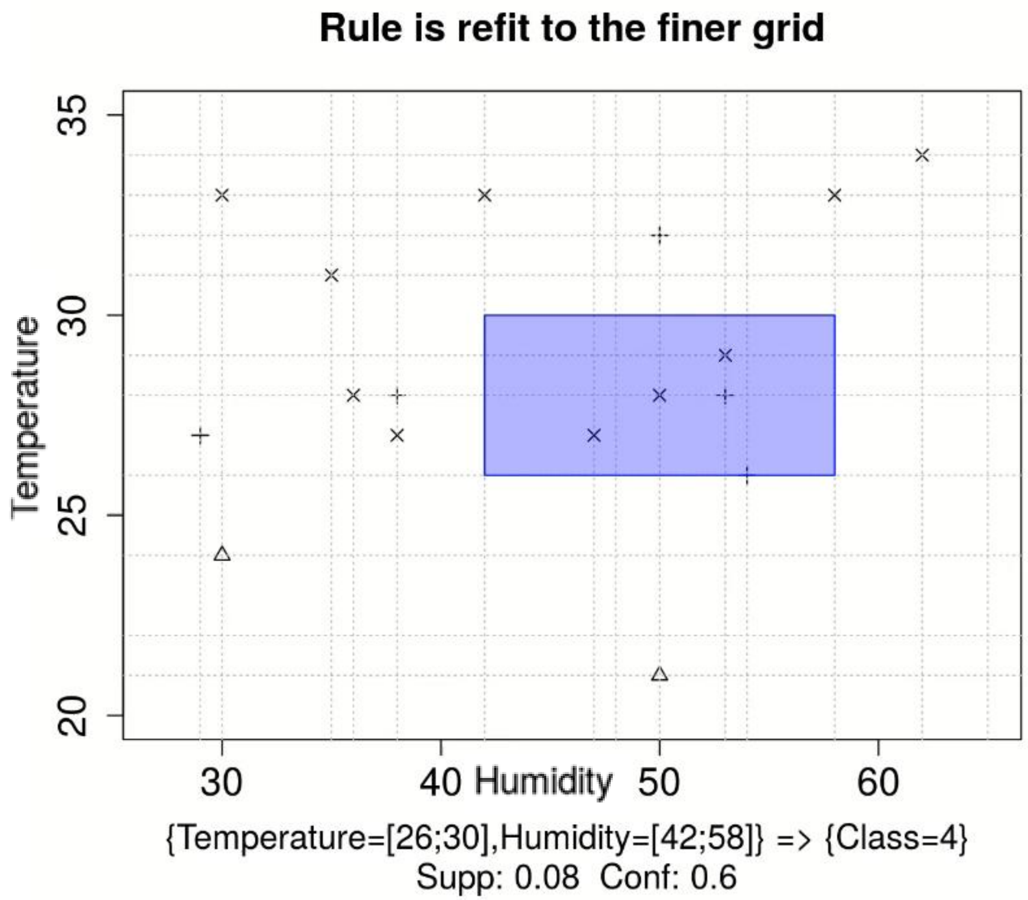
Trimming has been applied and the rule is shaved of boundary regions that do not cover any correctly classified instances: one instance with class 3 (denoted by a red box) on the rule boundary is initially covered by Rule #3, but also misclassified by it. As part of the trimming, the lower boundary of the Temperature literal on Rule #3 is increased not to cover this instance.

QCBA: Extension

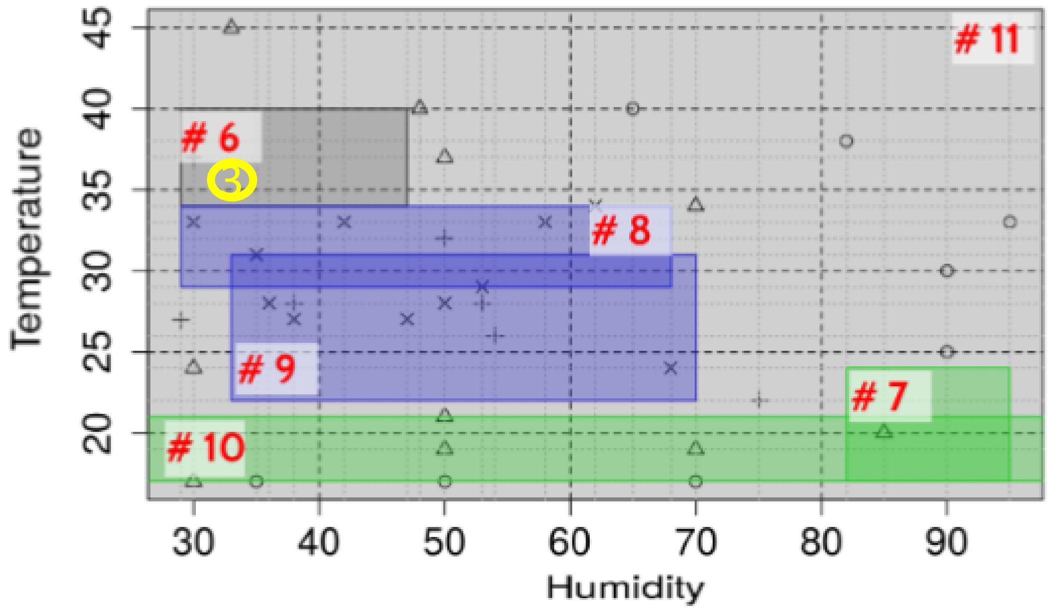
The purpose of this hill-climbing process is to move literal boundaries outwards, increasing rule coverage by extending the boundaries of literals in its antecedent in one direction with steps corresponding to breakpoints on the finer grid.

An extension is accepted if there is an improvement in confidence over the last confirmed extension by a user-set minImprovement threshold.

If this crisp extension has not been reached, a conditional extension is attempted to overcome a temporal decrease in the confidence of the enlarged rule.



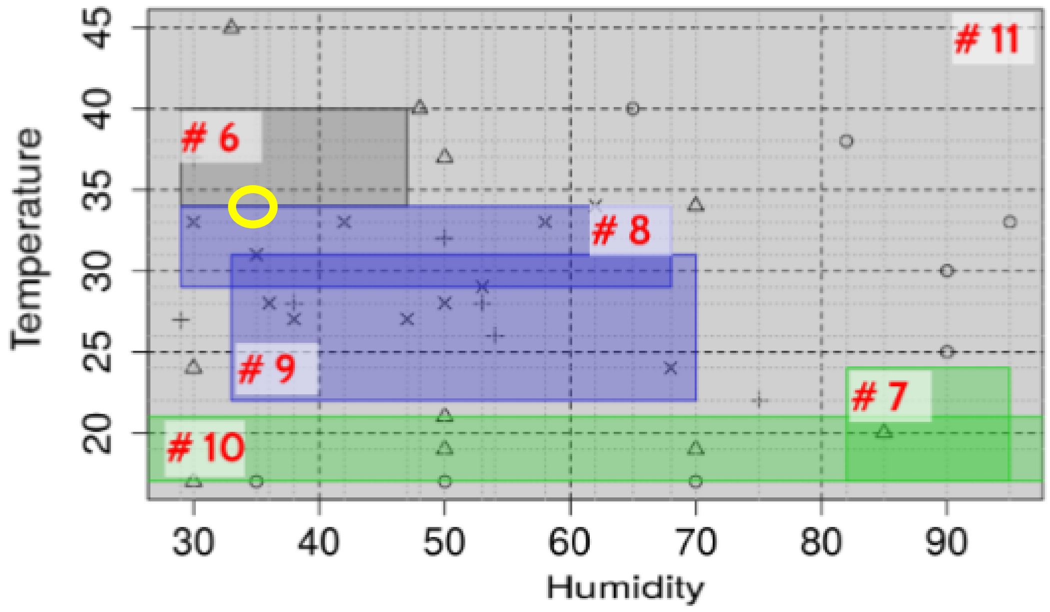
QCBA: Default overlap pruning – **instance** based




```
# 6   {Temperature=[34;40],Humidity=[29;47]} => {Class=3}
# 7   {Temperature=[17;24],Humidity=[82;95]} => {Class=2}
# 8   {Temperature=[29;34],Humidity=[29;68]} => {Class=4}
# 9   {Temperature=[22;31],Humidity=[33;70]} => {Class=4}
# 10                  {Temperature=[17;21]} => {Class=2}
# 11                                     {} => {Class=3}
```

Rule #6 is the only pruning candidate since other rules classify to different classes than the default rule (Rule #11). Because none of the rules between #6 and #11 would cause misclassification of training instances covered by #6 if #6 is removed, Rule #6 is removed by instance-based pruning.

QCBA: Default overlap pruning – **range** based




```
# 6   {Temperature=[34;40],Humidity=[29;47]} => {Class=3}
# 7   {Temperature=[17;24],Humidity=[82;95]} => {Class=2}
# 8   {Temperature=[29;34],Humidity=[29;68]} => {Class=4}
# 9   {Temperature=[22;31],Humidity=[33;70]} => {Class=4}
# 10                   {Temperature=[17;21]} => {Class=2}
# 11                                      {} => {Class=3}
```

Rules between #6 and #11 seem to cover different geometric regions, which would be an argument for removing #6. After closer inspection (Fig. 2 right) we note that #6 overlaps in Humidity and shares an inclusive boundary on Temperature with Rule #8, which also includes Temperature=34 but classifies to a different class, therefore it is not removed by range-based pruning. If this rule was removed, a hypothetical test instance (in yellow) would be classified into class 4 by Rule #8 instead of to class 3 by #6 (or the default rule).

Experiments

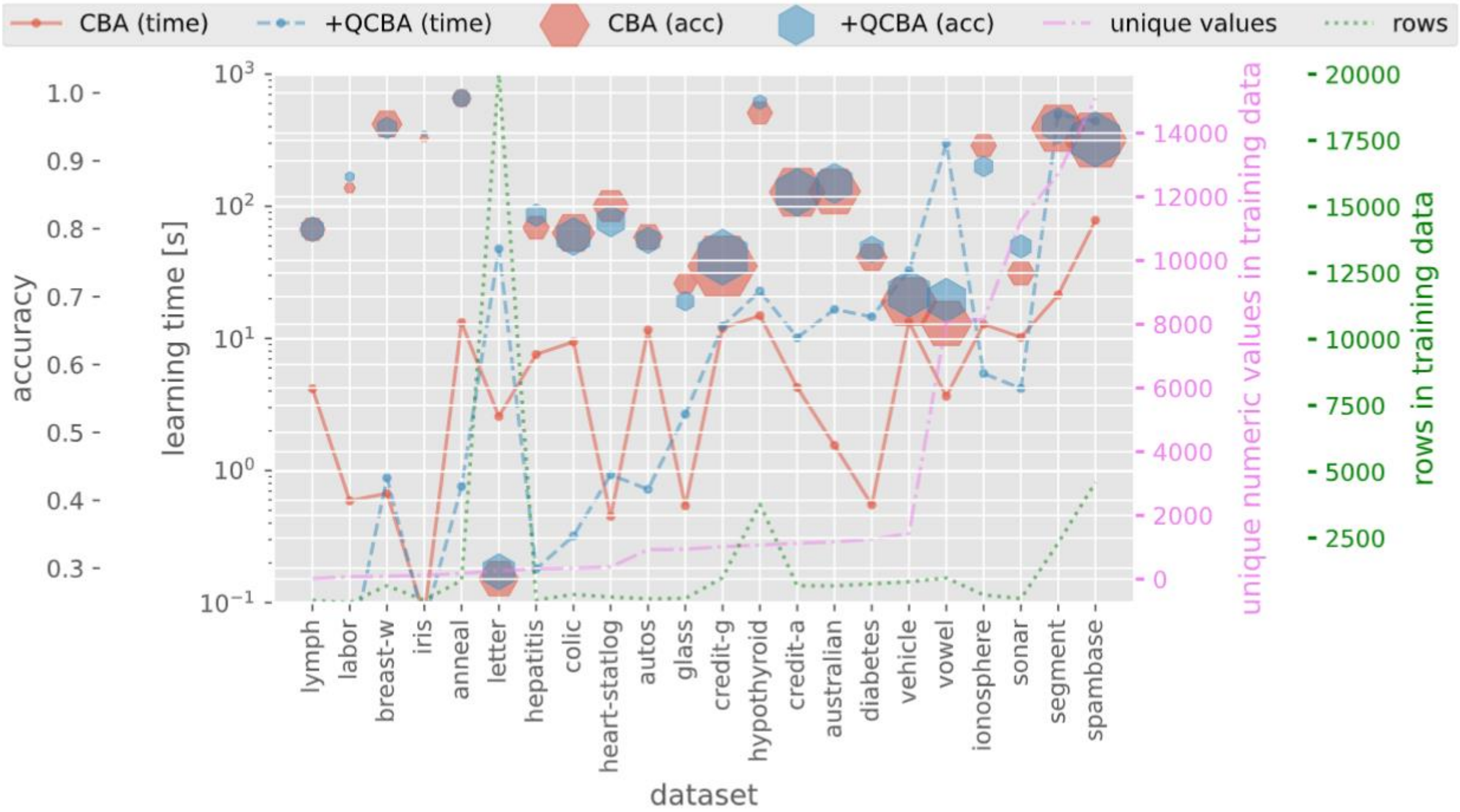
dataset	att.	inst.	miss.	class	description
anneal	39	898	Y	nominal (6)	steel annealing dataset
australian	15	690	N	binary	credit card applications
autos	26	205	Y	ordinal (7)	riskiness of second hand cars
breast-w	10	699	Y	binary	breast cancer
colic	23	368	Y	binary	horse colic (surgical or not)
credit-a	16	690	Y	binary	credit approval
credit-g	21	1000	N	binary	credit risk
diabetes	9	768	N	binary	diabetes
glass	10	214	N	nominal (6)	types of glass
heart-statlog	14	270	N	binary	diagnosis of heart disease
hepatitis	20	155	Y	binary	hepatitis prognosis (die/live)
hypothyroid	30	3772	Y	nominal (3)	thyroid disease data set
ionosphere	35	351	N	binary	radar data
iris	5	150	N	nominal (3)	iris (flower) varieties
labor	17	57	Y	ordinal (3)	contributions to health plan
letter	17	20000	N	nominal (26)	letter recognition
lymph	19	148	N	nominal (4)	lymphography domain
segment	20	2310	N	nominal (7)	image segment classification
sonar	61	208	N	binary	object based on sonar signal
spambase	58	4601	N	binary	spam detection
vehicle	19	846	N	nominal (4)	object type based on silhouette
vowel	13	990	N	nominal (11)	vowel recognition

+ KDD’99 Anomaly detection datased

Experiments – ablation study

configuration	cba	#1	#2	#3	#4	#5	#6	#7
CBA as a baseline and a source of input candidate rules for postprocessing								
data coverage pruning	Y	Y	Y	Y	Y	Y	Y	Y
default rule pruning	Y	-	-	-	-	-	-	-
QCBA postprocessing steps								
refit	na	Y	Y	Y	Y	Y	Y	Y
literal pruning	na	-	Y	Y	Y	Y	Y	Y
trimming	na	-	-	Y	Y	Y	Y	Y
extension	na	-	-	-	Y	Y	Y	Y
post-pruning – data coverage	na	-	-	-	-	Y	Y	Y
post-pruning – default rule	na	-	-	-	-	Y	Y	Y
def. rule overlap – instance	na	-	-	-	-	-	Y	-
def. rule overlap – range	na	-	-	-	-	-	-	Y
Results								
won/tie/loss (QCBA vs base)	base	9/6/7	9/6/7	7/7/8	8/7/7	10/9/3	7/6/9	10/9/3
p-value (QCBA vs base)	base	0.58	0.58	0.62	0.82	0.23	0.09	0.23
accuracy	0.81	0.81	0.81	0.81	0.81	0.81	0.80	0.81
avg AUC for binary class datasets	0.82	0.82	0.82	0.83	0.82	0.82	0.80	0.82
avg conditions / rule	2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8
avg number of rules	84.2	91.8	91.8	91.8	91.8	65.9	47.8	65.5
avg conditions / model	235.3	258.7	258.7	258.7	258.7	183.6	132.0	182.4
Median learning time [s]	6.2	0.2	0.2	0.3	4.7	4.8	5.0	5.2
– Normalized	1.00	0.02	0.04	0.05	0.76	0.77	0.81	0.83

Experiments – individual datasets (QCBA5)



Experiments – postprocessing other models

	only CMAR	+Q#5	only CPAR	+Q#5	only FOIL2	+Q#5	only PRM	+Q#5
accuracy	0.83	0.83	0.81	0.83	0.82	0.84	0.81	0.83
won/tie/loss (base+QCBA vs base)	base	10/2/10	base	15/5/2	base	14/5/3	base	14/4/4
p-value (base+QCBA vs base)	base	0.94	base	0.00	base	0.02	base	0.01
avg number of rules	489.2	112.7	88.5	61.9	107.1	76.9	80.1	61.5
avg conditions / rule	3.0	2.7	2.0	2.0	2.5	2.3	2.0	2.0
avg conditions / model	1462.0	302.6	178.9	126.7	263.5	176.7	161.4	125.4
median learning time [s]	2.8	13.4	0.3	0.7	0.3	1.1	0.3	0.7
- normalized	1.00	4.85	1.00	2.55	1.00	3.81	1.00	2.62

Li, W., Han, J., Pei, J.: CMAR: Accurate and efficient classification based on multiple class-association rules. In: Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on, pp. 369–376. IEEE (2001)

Yin, X., Han, J.: CPAR: Classification based on predictive association rules. In: Proceedings of the 2003 SIAM International Conference on Data Mining, pp. 331–335. SIAM (2003)

Quinlan, J.R., Cameron-Jones, R.M.: FOIL: A midterm report. In: European conference on machine learning, pp. 1–20. Springer (1993)

Using LUCS-KDD Software Library Java implementations available via arulesCBA

Experiments – postprocessing IDS models

literal pruning	only IDS NA	QCBA (+Q#5) -	QCBA (+Q#5) Y	QCBA (+Q#6) -	QCBA (+Q#6) Y
accuracy	0.61	0.63	0.63	0.63	0.63
won/tie/loss (base+QCBA vs base)	base	14/7/1	13/8/1	13/8/1	12/9/1
p-value (base+QCBA vs base)	base	0.00	0.00	0.00	0.00
avg number of rules	16.6	4.6	4.2	3.7	3.4
avg conditions / rule	3.6	2.1	1.7	1.9	1.5
avg conditions / model	60.4	9.6	7.1	7.2	5.3
median learning time [s]	21.1	2.8	1.3	2.8	1.5
- normalized	1.00	0.13	0.06	0.13	0.07

Lakkaraju, H., Bach, S.H., Leskovec, J.: Interpretable decision sets: A joint framework for description and prediction. In: Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, pp. 1675–1684. ACM, New York, NY, USA (2016)

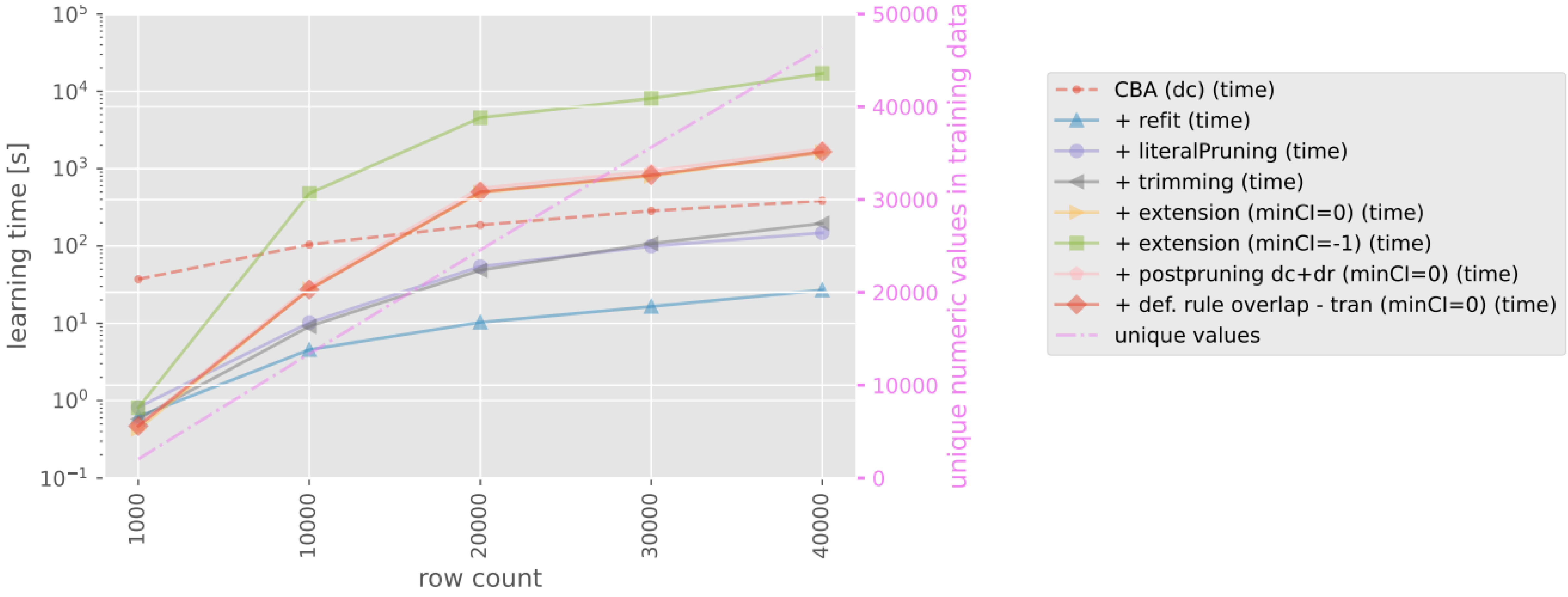
Using own implementation: Filip, Jiri, and Tomas Kliegr. "PyIDS-Python Implementation of Interpretable Decision Sets Algorithm by Lakkaraju et al, 2016." RuleML+ RR (Supplement). 2019.

Experiments – postprocessing SBRL models

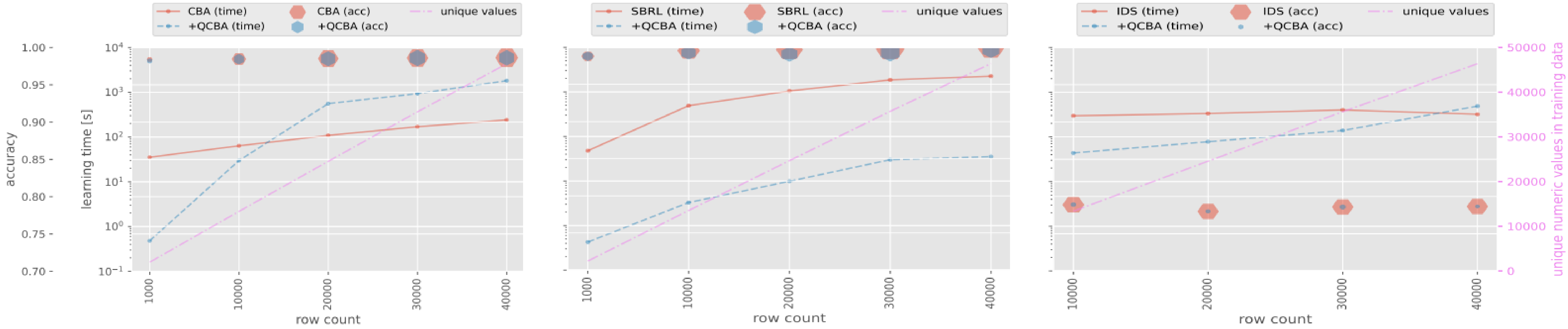
	SBRL trained on short rules			SBRL trained on long rules		
	only SBRL	QCBA (+Q#5)	QCBA (+Q#6)	only SBRL	QCBA (+Q#5)	QCBA (+Q#6)
accuracy	0.80	0.81	0.79	0.81	0.81	0.80
won/tie/loss (base+QCBA vs base)	base	6-5-0	2-5-4	base	7-1-3	6-1-4
p-value (base+QCBA vs base)	base	0.03	0.25	base	0.75	0.72
avg number of rules	4.8	3.7	3.4	3.3	3	2.7
avg conditions / rule	0.8	0.7	0.6	1.5	1.2	1.2
avg conditions / model	3.7	2.5	2.2	4.8	3.6	3.2
median learning time [s]	0.6	0.1	0.1	24.9	0.1	0.1
- normalized	1.00	0.17	0.17	1.00	0	0

Yang, H., Rudin, C., Seltzer, M.: Scalable bayesian rule lists. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 3921–3930. JMLR (2017)

Experiments – scalability



Experiments – scalability



(a) CBA+QCBA (b) SBRL+QCBA (c) IDS+QCBA

Experiments – other baselines

reference classifier	CORELS	J48	PART	RIPPER	FURIA
accuracy	0.81	0.84	0.85	0.79	0.85
won/tie/loss (CBA+QCBA vs reference)	9/1/1	12/2/8	10/3/8	12/5/5	6/4/11
p-value (CBA+QCBA vs reference)	0.05	0.40	0.51	0.12	0.22

CORELS: *Angelino, E., Larus-Stone, N., Alabi, D., Seltzer, M., Rudin, C.: Learning certifiably optimal rule lists for categorical data. The Journal of Machine Learning Research 18(1), 8753–8830 (2017)*

J48: *Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)*

PART: *Cohen, W.W.: Fast effective rule induction. In: Proceedings of the Twelfth International Conference on International Conference on Machine Learning, ICML'95, pp. 115–123. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1995)*

RIPPER: *Frank, E., Witten, I.H.: Generating accurate rule sets without global optimization. In: Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98, p. 144–151. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1998)*

FURIA: *Hühn, J., Hüllermeier, E.: FURIA: an algorithm for unordered fuzzy rule induction. Data Mining and Knowledge Discovery 19(3), 293–319 (2009)*

Experiments – per dataset results + additional baselines

dataset	wins of base+QCBA over base 1	2	3	4	5	6	7	wins of CBA+QCBA over reference 8	9	10	11	12	QCBA % wins	best result for dataset accuracy	algorithm
labor	1	1	1	1	1	1	NA	1	NA	1	1	1	100	0.927	**CPAR+QCBA**
australian	1	0	0	1	1	0	1	1	1	1	1	1	75	0.868	CMAR
credit-g	0	1	1	0	1	1	0	1	1	1	1	1	75	0.762	CBA
letter	1	1	1	1	1	1	NA	NA	NA	0	NA	0	75	0.879	J48
vehicle	1	1	1	1	1	1	NA	0	NA	0	0	1	70	0.729	PART
diabetes	1	1	1	0	0	0	0	1	1	1	1	1	67	0.770	**CBA+QCBA**
hepatitis	1	0	1	0	0	0	1	1	1	1	1	1	67	0.832	CMAR
sonar	1	0	1	1	0	1	0	0	1	1	1	1	67	0.793	FURIA
iris	1	1	0	1	1	0	NA	1	NA	0	1	0	60	0.960	**CMAR+QCBA**
lymph	0	0	0	1	1	1	NA	0	NA	1	1	1	60	0.866	FURIA
vowel	1	1	1	1	1	1	NA	0	NA	0	0	0	60	0.833	J48
breast-w	0	0	1	1	1	1	1	0	1	1	0	0	58	0.969	CMAR
autos	0	0	1	1	1	1	NA	0	NA	0	0	1	50	0.819	**FOIL2+QCBA**
glass	0	0	1	0	1	1	NA	0	NA	1	0	1	50	0.724	CBA
hypothyroid	1	1	1	1	0	1	NA	0	NA	0	0	0	50	0.996	J48
heart-statlog	0	0	0	1	0	0	1	0	1	1	0	1	42	0.833	FOIL2
ionosphere	0	0	0	0	1	0	1	0	1	1	1	0	42	0.929	PRM
anneal	0	0	1	0	0	0	NA	0	NA	1	1	1	40	0.993	**FOIL2+QCBA**
segment	0	1	1	0	1	1	NA	0	NA	0	0	0	40	0.969	FURIA
spambase	0	1	0	0	1	1	0	0	1	0	0	0	33	0.937	PART
colic	0	0	1	0	0	1	1	0	0	0	0	0	25	0.859	CORELS
credit-a	0	0	0	1	0	0	1	0	0	0	0	0	17	0.867	CMAR

Code availability

- arc (CRAN R Package with CBA implementation)
  - https://github.com/kliegr/arc
- QCBA (CRAN R Package with QCBA implementation)
  - Latest version with CPAR/CMAR/FOIL2/PRM support on Github
  - https://github.com/kliegr/QCBA
- pyARC (Python version of arc + QCBA implementation)
  - https://github.com/jirifilip/pyARC
- Benchmarking code
  - https://github.com/kliegr/arcBench

```
library(qCBA)

allData <-
datasets::iris[sample(nrow(datasets::iris)),]
trainFold <- allData[1:100,]
rmCBA <- cba(trainFold, classAtt="Species")
rmqCBA <- qcba(cbaRuleModel=rmCBA,datadf=trainFold)
print(rmqCBA@rules)
```

Papers

Kliegr, T., & Izquierdo, E. (2023). QCBA: improving rule classifiers learned from quantitative data by recovering information lost by discretisation. *Applied Intelligence*, 1-31.

Hahsler, M., Johnson, I., Kliegr, T., & Kuchař, J. (2019). Associative Classification in R: arc, arulesCBA, and rCBA. *R Journal*, *9*(2).