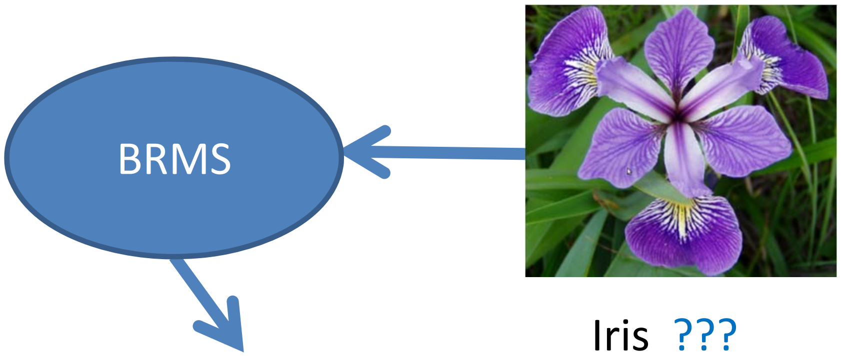
Learning Business Rules
with Association Rule Classifiers

Tomáš Kliegr[1,4], Jaroslav Kuchař[1,2], Davide Sottara[3], Stanislav Vojíř[1]

[1] Dep. of Inf. And Knowl. Eng., University of Economics, Prague
[2] Web Engineering Group, Czech Technical University
[3] Biomedical Informatics Department, Arizona State University
[4] Multimedia and Vision Research Group, Queen Mary, University of London

Motivation example

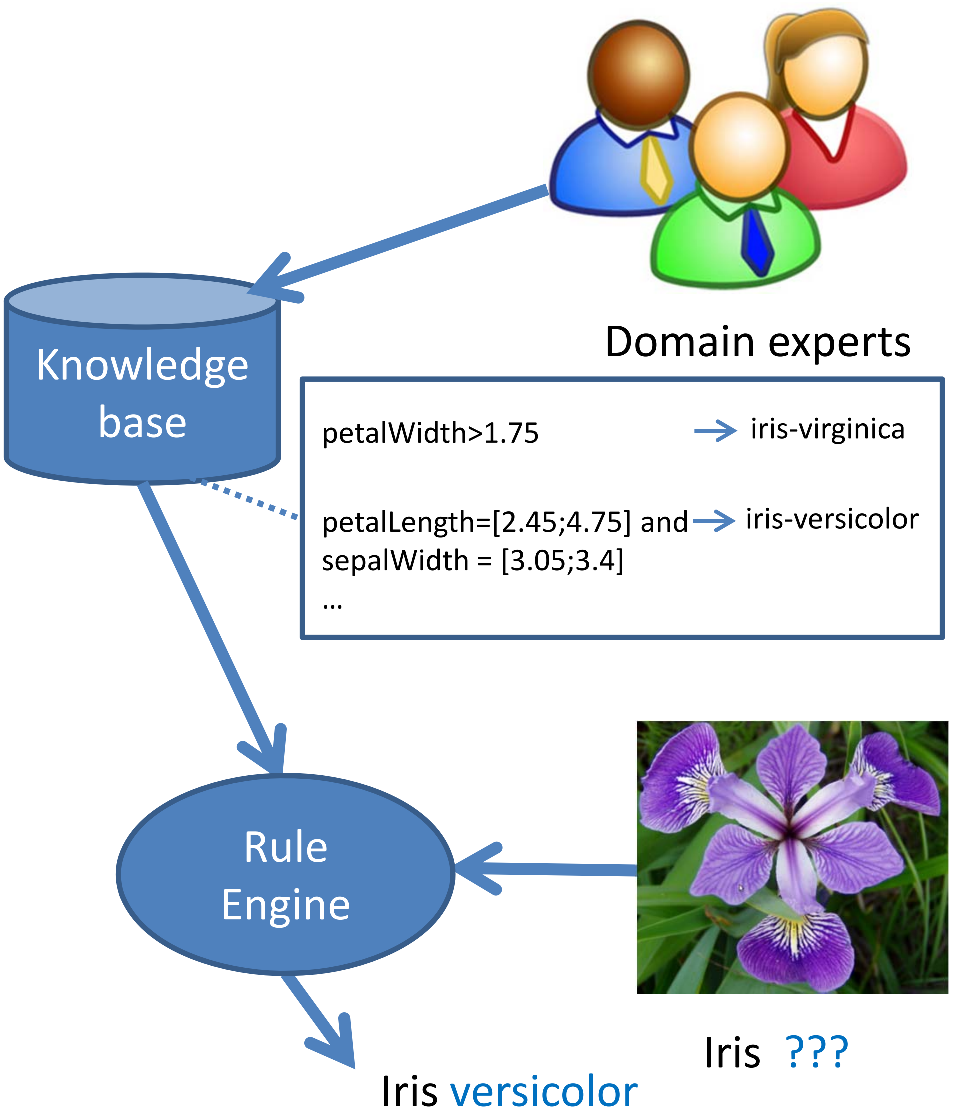


Iris ???

Iris versicolor

RESTRICTION: In this paper, we focus on "classification business rules".

Problem statement



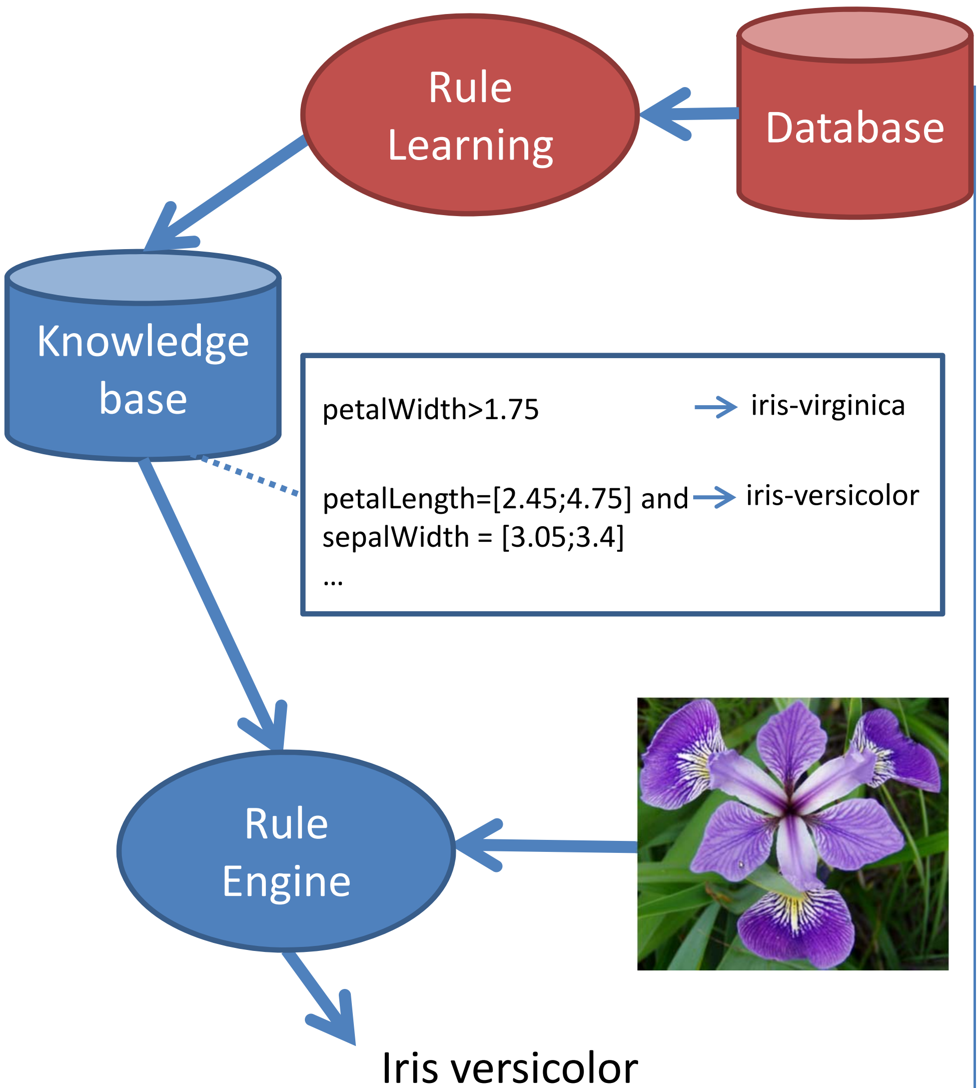
With Business Rule Management System (BRMS) applications can invoke decision logic which is input in the form of rules, instead of procedural code

\+ This reduces reliance on the IT experts

- Requires extensive subject matter expertise
- (A lot of) Expert time

RESTRICTION: In this paper, we focus on “classification business rules”.

Problem statement



Ideally, the rule learning algorithm executed on the database of iris varieties would substitute the human expert.

As we will see, rule learning algorithms often yields rule sets that are

- Conflicting
- Contain redundant rules
- Excessive number of rules
- Syntactically simple
- Probabilistic

Problem statement

- Conflicting
- Contain redundant rules
- Excessive number of rules
- Syntactically simple
- Probabilistic

R1: petalWidth>1.75 → iris-virginica, supp= 0.296, conf=1

R2: petalWidth>1.75 and sepalWidth = [3.05;3.4] → iris-virginica supp= 0.100, conf=1

…

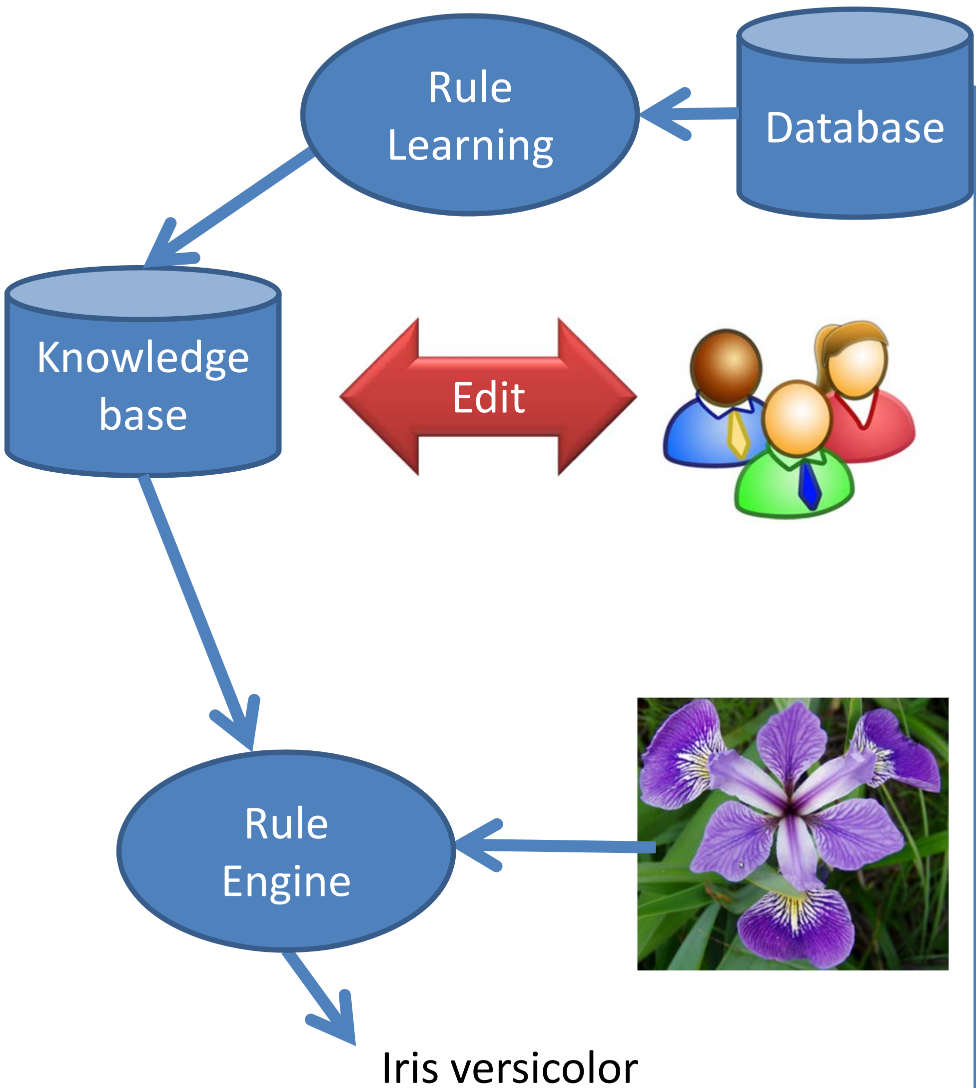
R9: sepalLength= (5.55;3.40] and sepalWidth<3.05 → iris-versicolor supp=0.230, conf=0.05

… 50 more rules

While this is not an issue for a completely automated "black box" classifier, in a business setting the policy can be that the rule set

a) is expert-reviewed before deployment,
b) each decision made by the system can be explained,
c) the rules must be convertible to a form that can be processes by BRMS

BR Learning Requirements



Business rule learning needs a rule-learning approach, which has

- BRMS supported rule expressiveness
- Syntactically rich
- Small number of output rules
- Exhaustive set of rules
- Ability to control rule quality

BRMS can then take care of

- Refine the rule base (by Subject Matter Expert)
- Execute rules
  - Classify objects at run time
  - Evaluate complex criteria
  - Handle uncertainty
- Manage rule conflicts
  - Defeasible logic, higher order rules, …

Choosing "base" classifier

- Multiple rule learning algorithms have been proposed.
- Focus on algorithms which match our criteria, but are additionally scientifically well established and with **tried open implementations**.

Two algorithms were shortlisted:

- RIPPER (Cohen 1995): 3017 citations in Google scholar. Available in the open source WEKA and RapidMiner systems.
- CBA (Liu et al., 98): 1968 citations GS. Based on apriori, which is available in most data mining systems.

Business rule learning needs a rule-learning approach, which has

- BRMS supported rule expressiveness
- Small number of output rules
- Exhaustive set of rules
- Expressive rule language
- Rule conflict resolution
- Ability to control rule quality

Eventually, we have settled for CBA

- in our experiments processing large datasets (N>10k) with RIPPER was unfeasible (JRIP - WEKA and RapidMiner impl.).
- CBA better fits the BR learning usecase
  - Better rule quality control
  - Exhaustive searchh

CLEF NewsReel'14
21.000 instances
1671 class labels

Classification based on Association Rules

- CBA generally proceeds as follows:
  1. Rule Generator
     - Mining of Class Association Rules based on Apriori
  2. Classifier Builder
     - v1 – many passes over the data
       1. Sort Rules (conf, supp, earlier)
       2. Pruning
          - For each rule iterate over all data
     - v2 – find best rule for each data case
       1. Sort Rules
       2. Preselecting of rules based on precedence
       3. Add candidate rules filtered out due to the lower precedence, improve coverage
       4. Final filter by total error + default class

Business rule learning needs a rule-learning approach, which has

- BRMS supported rule expressiveness
- Small number of output rules
- Exhaustive set of rules
- Expressive rule language
- Rule conflict resolution
- Ability to control rule quality

Business rule CBA (brCBA)

CBA (Liu et al., 98) nearly matches the requirements of BR learning

Why brCBA?

- CBA has three rule pruning steps
  - Difficult to track why a specific rule was removed

Business rule learning needs a rule-learning approach, which has

- BRMS supported rule expressiveness
- Small number of output rules
- Exhaustive set of rules
- Expressive rule language
- Rule conflict resolution
- Ability to control rule quality

Why brCBA?

- CBA has three rule pruning steps
  - Difficult to track why a specific rule was removed
- CBA learns conjunctive rules

Business rule learning needs a rule-learning approach, which has

- BRMS supported rule expressiveness
- Small number of output rules
- Exhaustive set of rules
- Expressive rule language
- Rule conflict resolution
- Ability to control rule quality

Why brCBA?

- CBA has three rule pruning steps
  - Difficult to track why a specific rule was removed
- CBA learns conjunctive rules
- CBA is a complete classifier
  - Adds unnecessary complexity when data is not separable
  - The default rule pruning step may deteriorate overall rule set accuracy far below the preset confidence threshold
- The default rule forces all objects into one of the target classes, which is undesirable in many business rule use cases

Business rule learning needs a rule-learning approach, which has

- BRMS supported rule expressiveness
- Small number of output rules
- Exhaustive set of rules
- Expressive rule language
- Rule conflict resolution
- Ability to control rule quality

brCBA

- brCBA is a simplification of CBA, so that the algorithm can be quickly built on top of standard association rule learning implementation (e.g. Christian Borgelt's arules package in R).

Rule learning (brCBA)

1. Learn association rules (constrained to contain the class attribute in consequent) with GUHA method
2. Perform data coverage pruning

Classification (same as in CBA algorithm)

A standard BRMS rule engine can be used to apply the model (rule set) on data

Business rule learning needs a rule-learning approach, which has

- BRMS supported rule expressiveness
- Small number of output rules
- Exhaustive set of rules
- Expressive rule language
- Rule conflict resolution
- Ability to control rule quality

brCBA

- brCBA is a simplification of CBA, so that the algorithm can be quickly built on top of standard association rule learning implementation (e.g. Christian Borgelt's arules package in R)

Rule learning (brCBA)

1. Learn association rules (constrained to contain the class attribute in consequent) with GUHA method
2. Perform data coverage pruning

Classification (same as in CBA algorithm)

A standard BRMS rule engine can be used to apply the model (rule set) on data

Business rule learning needs a rule-learning approach, which has

- BRMS supported rule expressiveness
- Small number of output rules
- Exhaustive set of rules
- Expressive rule language
- Rule conflict resolution
- Ability to control rule quality

- The data coverage pruning makes it simple to understand for business analyst why a specific rule output in the association rule learning was removed. No other pruning is performed.

brCBA

- brCBA is a simplification of CBA, so that the algorithm can be quickly built on top of standard association rule learning implementation (e.g. Christian Borgelt's arules package in R)

Rule learning (brCBA)

1. Learn association rules (constrained to contain the class attribute in consequent) with GUHA Method
2. Perform data coverage pruning

Classification (same as in CBA algorithm)

A standard BRMS rule engine can be used to apply the model (rule set) on data

Business rule learning needs a rule-learning approach, which has

- BRMS supported rule expressiveness
- Small number of output rules
- Exhaustive set of rules
- Expressive rule language
- Rule conflict resolution
- Ability to control rule quality

- The data coverage pruning makes it simple to understand for business analyst why a specific rule output in the association rule learning was removed. No other pruning is performed.
- The absence of default rule pruning ensures that all rules matching the specified quality measures (minSupp and minConf) are on the output, and no default rule of possibly lower quality is introduced by the algorithm

brCBA

- brCBA is a simplification of CBA, so that the algorithm can be quickly built on top of standard association rule learning implementation (e.g. Christian Borgelt's arules package in R)

Rule learning (brCBA)

1. Learn association rules (constrained to contain the class attribute in consequent) with GUHA Method
2. Perform data coverage pruning

Classification (same as in CBA algorithm)

A standard BRMS rule engine can be used to apply the model (rule set) on data

Business rule learning needs a rule-learning approach, which has

- BRMS supported rule expressiveness
- Small number of output rules
- Exhaustive set of rules
- Expressive rule language
- Rule conflict resolution
- Ability to control rule quality

- The data coverage pruning makes it simple to understand for business analyst why a specific rule output in the association rule learning was removed. No other pruning is performed.
- The absence of default rule pruning ensures that all rules matching the specified quality measures (minSupp and minConf) are on the output, and no default rule of possibly lower quality is introduced by the algorithm
- GUHA method learns rich association rules with disjunctions and negations

Rule Pruning

- Data coverage pruning is the most commonly used pruning technique in CBA-derived algorithms

Algorithm 1 Data Coverage

Require: rules – sorted list of rules, T – set of objects in the training dataset
Ensure: rules – pruned list of rules

```
rules := sort rules according to criteria
for all rule ∈ rules do
    matches:= set of objects from T that match both rule ant. and conseq.
    if matches==∅ then
        remove rule from rules
    else
        remove matches from T
    end if
end for
return rules
```

Rule ranking criteria

- Confidence
- Support
- Rule length (shorter is better)

Experiment objectives

- Evaluate impact of pruning
  - No pruning (use apriori output directly for classification)
  - brCBA (apriori, then data coverage pruning)
  - Original CBA (data coverage, pessimistic and default rule pruning)
- Evaluate the impact and sensitivity to:
  - minSupport threshold
  - minConfidence threshold
- Evaluate the impact of added rule language expressivity
  - negations
  - disjunctions in rule body

Experimental setup

Datasets

- UCI: Iris, Glass

Dataset	Rows	Attributes
Iris	150	4
Glass	214	9

Preprocessing

- Numerical attributes were discretized with equidistant binning with custom merging of bins with small support

Rule learning

- LISp-Miner implementation, apriori-like setup

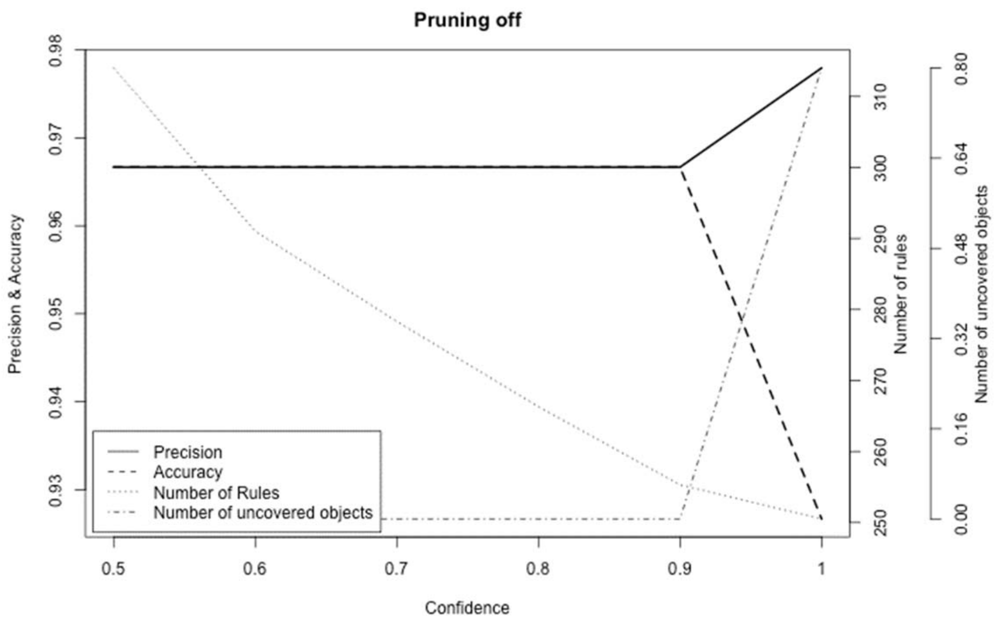
Pruning

- Data coverage pruning on/off

<u>Experiment objectives</u>

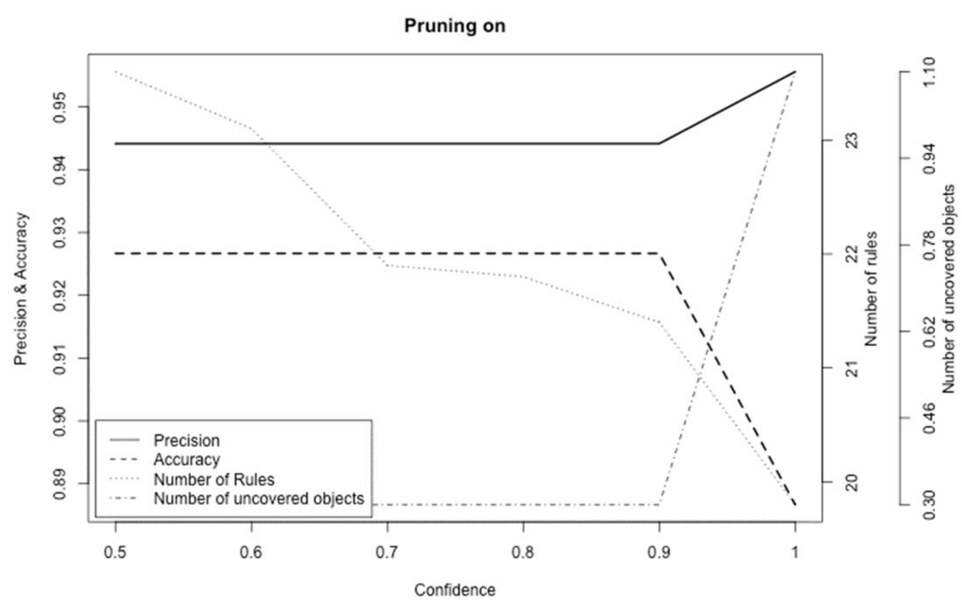
1) Compare results with other classifiers
2) Determine impact of:
- minSupport thr.
- minConfidence thr.
- pruning

Experimental results pruning



Effect of pruning. Iris dataset, minimum support threshold 1

iris dataset



Pruning: decreased the rule count by 90%, lowering accuracy only by 1%

+ for business rule learning

Experimental results minSupp

Dataset, task	support	not pruned Rules	not pruned Accuracy	pruned Rules	pruned Accuracy
iris	10	87	0.940	19	0.920
”	2	168	0.947	21	0.913
”	1	291	**0.967**	23	0.927
iris, sequence 1-2	10	904	0.940	17	0.953
”	2	1661	0.953	19	**0.960**
”	1	2653	**0.960**	19	**0.960**
glass	10	32	0.464	21	0.464
”	2	2374	**0.622**	68	0.608
balance scale	10	124	**0.891**	78	0.870
”	2	558	0.841	216	0.714
balance scale, subset 1-2	10	11947	0.758	153	**0.779**

Impact of minimum support threshold, minConf=0.6

Support: The lower, the better (and slower).

Experimental results minConf

confidence	not pruned Rules	Accuracy	pruned Rules	Accuracy
0.5	58.3	0.529	25.8	**0.534**
0.6	31.8	0.464	21.1	0.464
0.7	10.3	0.290	8.4	0.286
0.8	2.4	0.117	1.8	0.117
0.9	0.4	0.010	0.2	0.010

Glass, minSupp=10 objects (5.18%)

confidence	not pruned Rules	Accuracy	pruned Rules	Accuracy
0.5	96	0.940	20	0.920
0.6	87	0.940	19	0.920
0.7	83	0.940	17	0.920
0.8	76	0.940	17	0.920
0.9	68	0.900	15	0.880

Iris, minSupp=10 objects (1.78%)

confidence	not pruned Rules	Accuracy	pruned Rules	Accuracy
0.6	124	0.891	78	0.870
0.7	86	0.875	70	0.864
0.8	50	0.790	50	0.782
0.9	24	0.547	24	0.547
1.0	1	0.047	1	0.047

Balancescale, minSupp 10 objects (1.78%)

Confidence: The lower, the better.

Additional experiments

Datasets

- UCI: Iris, Balance scale, Glass

Preprocessing

- Numerical attributes were discretized with equidistant binning with custom merging of bins with small support

Dataset	Rows	Attributes	Bins after preprocessing
Iris	150	4	18
BalanceScale	625	4	20
Glass	214	9	19

Rule learning

- Default run (as in apriori)
- **Negations**
  - **for each item, a dual "negated" item is created**
- **Dynamic binning – nominal attributes ("subset" length = 2)**
- **Dynamic binning – cardinal attributes ("interval" length = 2)**

Pruning

- Data coverage pruning on/off

Higher expressivity rules with GUHA

- The standard apriori algorithm outputs **conjunctive** rules
- BRMS systems routinely work with rules that contain **disjunctions** between attribute values (**dynamic binning**) or **negated literals**.
- In our experiments, we have employed in the LISp-Miner system which unlike apriori implementations is able to learn higher expressiveness rules.

Original intervals
[20-25)
[25-30)
[30-35)
[35-40)
[40-45)
[45-50)

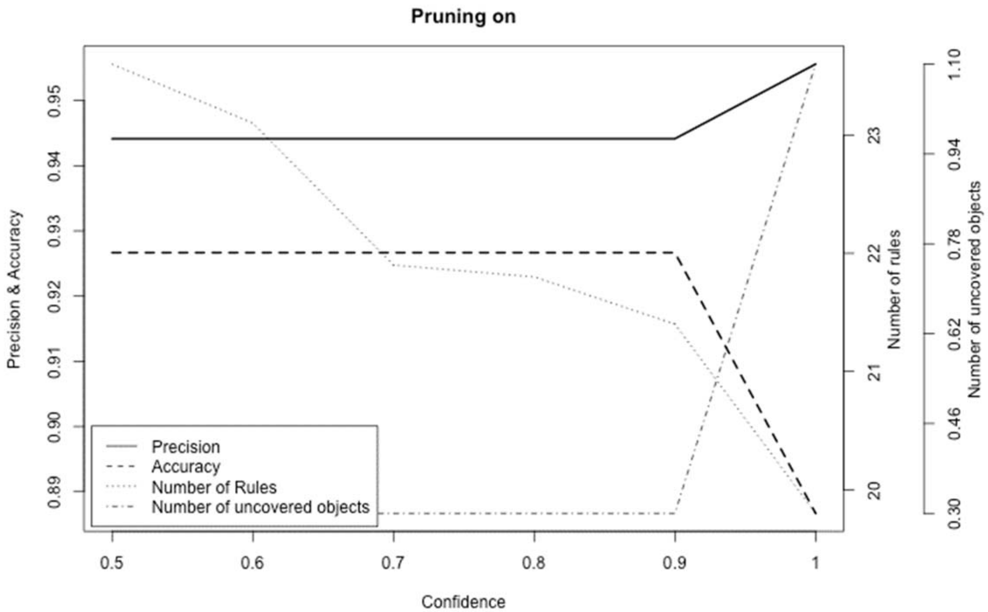
Sequence 1-2
Original intervals…
[20-25),[25-30)
[25-30),[30-35)
[30-35),[35-40)
[35-40),[40-45)
…

Subset 1-2
Original intervals…
[20-25),[25-30)
[20-25),[30-35)
[20-25),[35-40)
[20-25),[40-45)
[25-30),[30-35)
…

Sequence: *binning of following categories (for ordinal attributes)*
Subset: *binning of categories regardless of the order*
Length 1-2: *generated bins contain at least 1 and maximum 2 original bins*

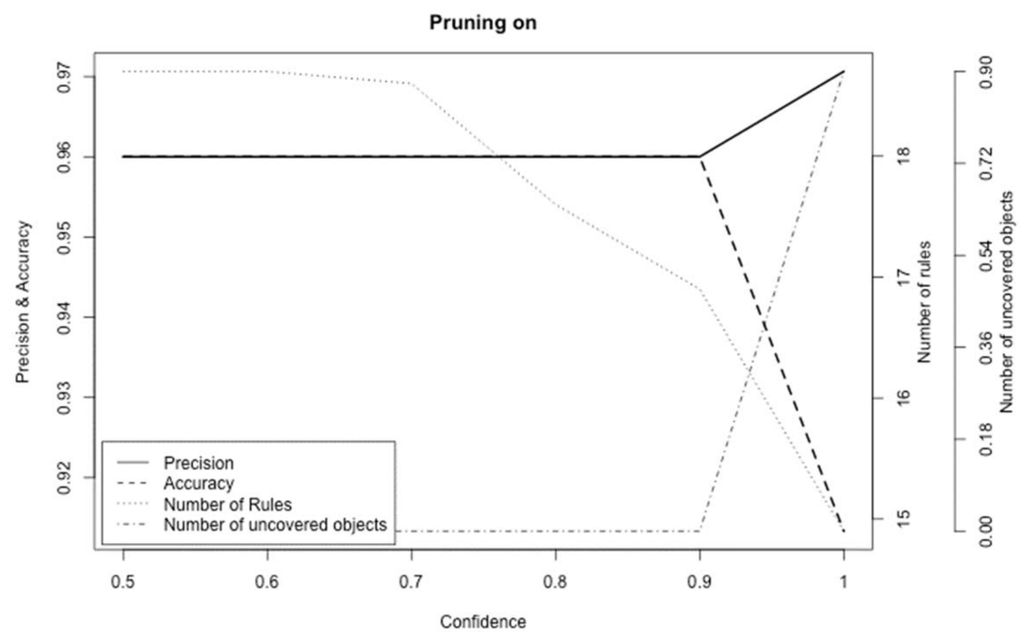
Experimental results dynamic binning

Effect of dynamic binning on cardinal attributes.
Iris dataset



Dynamic binning off

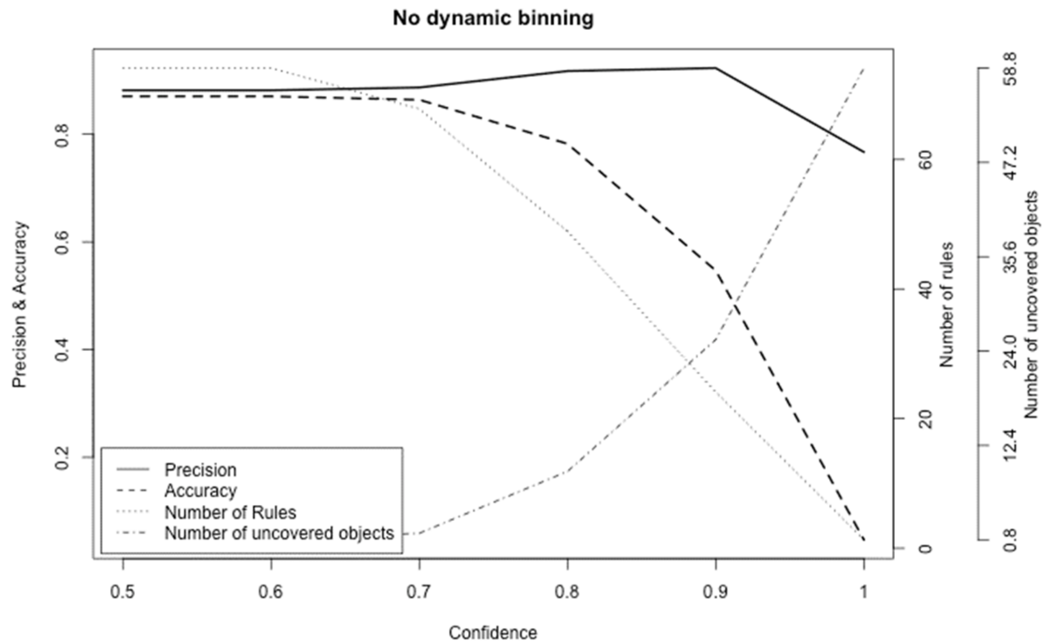
sepalWidth = [3.2;3.44) or sepalWidth = [3.44;3.68) => XClass(Iris-setosa)



Dynamic binning on

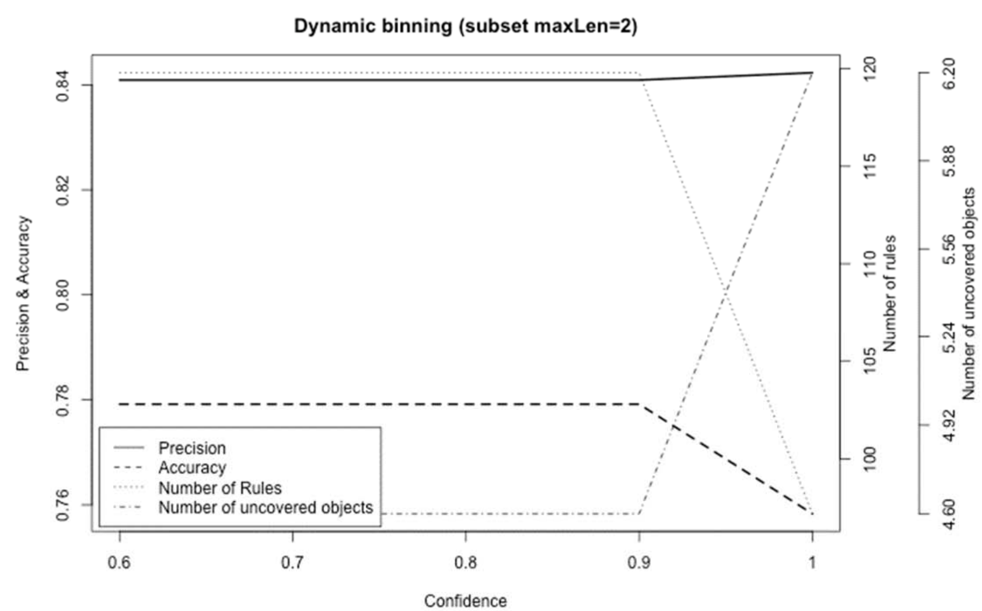
Dynamic binning (cardinal attributes) – better accuracy (3.4% improvement) and lower rule count (18 vs 23). However – **much** longer learning time.

Experimental results dynamic binning



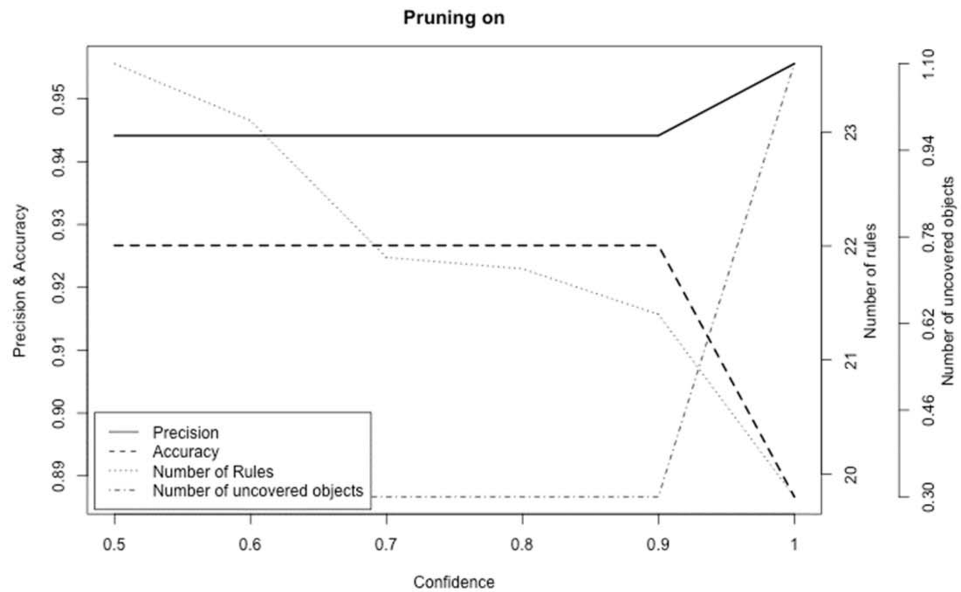
Effect of dynamic binning on nominal attributes.
Balancescale dataset

(LeftDistance=S or LeftDistance=M) and (LeftWeight=L or LeftWeight=H) => XClass=L



Dynamic binning (nominal attributes)– worse accuracy, higher rule count and **drastically** longer learning time.

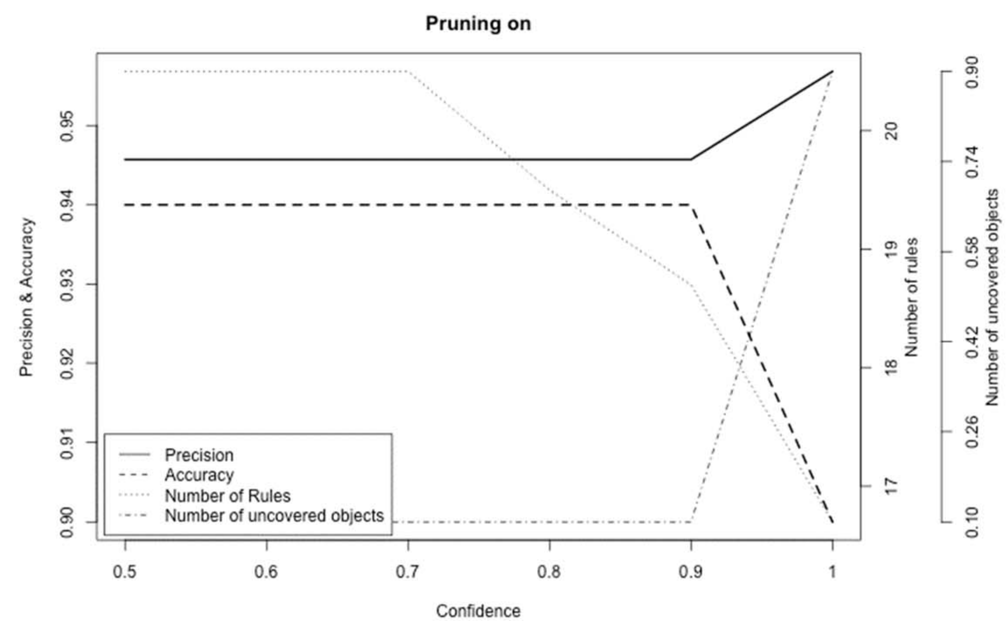
Experimental results negative literals



Effect of including negative literals.
Iris dataset

petalLength =[1;1.59)
and petalWidth=[0.1;0.34)
and not(sepalLength=[4.3;4.66)
and not(sepalWidth=[2;2.34)
=> XClass(Iris-setosa)

Negative literals – worse accuracy, higher rule count and higher learning time.



Experimental results time complexity

Dataset/Task	Attributes	Verifications	Rules	Mining duration
	without binning	315	80	less than 1 s
	with negations	13 542	2 472	12 s
	disjunctions (nominal)	19 413	4 715	27 s
BalanceScale (min Conf 0,5)	without binning	510	146	less than 1 s
	with negations	33 045	9 040	43 s
	disjunctions (nominal)	73 230	17 004	99 s
	disjunctions (cardinal)	9 582	2 122	10 s
	disjunctions (cardinal – 3 values)	45 915	11 846	75 s
Glass (min Conf 0,9)	without binning	3 920	24	less than 1 s
	with negations	669 075	8 146	64 s
	dynamic binning	*not suitable (attributes have only 2 values)*		

Experimental results overview

dataset	previously reported results C4.5	ripper	cba	cmar	cpar	brCBA not pruned	pruned
iris	0.953	0.940	0.947	0.940	0.947	**0.967**	0.960
glass	0.687	0.691	**0.739**	0.701	0.744	0.622	0.612

brCBA modification of CBA is not only simpler than CBA, but also fully matches our criteria for business rule learning (due to removal of default rule pruning).

Requirements

- BRMS supported rule expressiveness
- Small number of output rules
- Exhaustive set of rules
- Expressive rule language
- Rule conflict resolution
- Ability to control rule quality

brCBA

d

Redundant rules removed by pruning

GUHA rules – no improvement

Classification by highest ranked rule

All rules matching confidence and support thresholds

Future work

- Perform benchmarks on all 25 UCI datasets used in the original CBA paper
- A certain complication in using brCBA (or apriori-based algorithms) for business rule learning not addressed in this paper is the need to perform discretization of numerical attributes in the preprocessing phase.
  - Hand designed bins are laborious to produce and may negatively impact performance
  - Automatic binning (e.g. Entropy-based) produces unnatural output, and may not also provide the best results
- We are looking for an algorithm that would remove the need to perform discretization, or at least soften its impact on classification performance.

Demo

- Demo of a Drools-based with brCBA rule learner will be presented at the RuleML Challenge - **Today from 14:50 at this room**

> Stanislav Vojir, Premysl Vaclav Duben and Tomas Kliegr. Business Rule Learning with Interactive Selection of Association Rules. RuleML Challenge'14
> http://easyminer.eu

- Demo of a recommender system using brCBA will be presented at PAIS

> Tomas Kliegr, Jaroslav Kuchar. Orwellian Eye: Video Recommendation with Microsoft Kinect. PAIS'14
> http://inbeat.eu/demo/pais14/