

# Different rule set pruning approaches



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

**2 Papers with different aims, but working in the same field: selection and pruning rule-sets.**

## ➤ **Content**

- Introduction in both papers
- The papers
  - Introduction
  - Used stuff
  - Experiments
  - Results and Conclusion
  - Critique
- Similarities between the papers
- Questions
- Used references

**With only 30min time, I try to present as informal as possible. If formals needed, they can be found in the references.**



# The papers

---

## **A Comparison of Rule Sets Generated from Databases by Indisceribility Relation – A Rough Sets Approach**

And

## **In the paper Data-Driven Adaptive Selection of Rules Quality Measures for Improving the Rules Induction Algorithm**

### **Common Issues**

- › Inductive rule learners
- › Growing and Pruning rule-sets
- › Measure methods
- › Experiments: comparison of new idea, with known and well working stuff

# A Comparison of Rule Sets Generated from Databases by Indisceribility Relation – A Rough Sets Approach



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

## Abstract

- Two Methods for generating rough sets are compared
- A version of well known ROSETTA software
- The editor's own programm called (5)
- A database with data about researching leptin





# Introduction

---

„A significant advantage of methods that yield decision trees or rule sets is that the models are directly inspectable and interpretable, and the results of decisions are explainable“ [1]

**This is the main aim of this paper,  
improving this advantage**

- **Pruning rules for better readability**
- **Raising the generality of rules**



# The Rough Sets Theory basics

**Now follows an informell explanation how Rough Sets work.**  
(if not informell, then I could just write down the whole text-passage)

## Rough Sets Theory

**A machine learning method, which leads to rule-sets.**

- › Universe has objects
- › Objects are describable with a certain amount of information
- › Object's information are expressed in attributes
- › Approach to handle imperfect knowledge [2]
- › Tries to separate negative and positive examples in rules
- › Undecided examples and objects, those without rules can be treated with additional methods





# The Rough Sets Theory basics

---

## Attribute-types of rules

- **Condition-attributes**

Holding values or intervalls for decisions

- **Decision-attributes**

Holding values or intervalls to what a rule leads



# The Rough Sets Theory basics

## Pattern of the generated rules

RULE1 := IF (c1 and c2 and c5) THEN d

## Condition-attributes

range in rule

c1 = [1 ; 2]

c2 = 0

c5 = {1 ; 2; 4}

definition Range

c1 : [real+]

c2 : [real]

c5 : [0... ..n]

## Decision-attributes

range in rule

d = [1 ; 5]

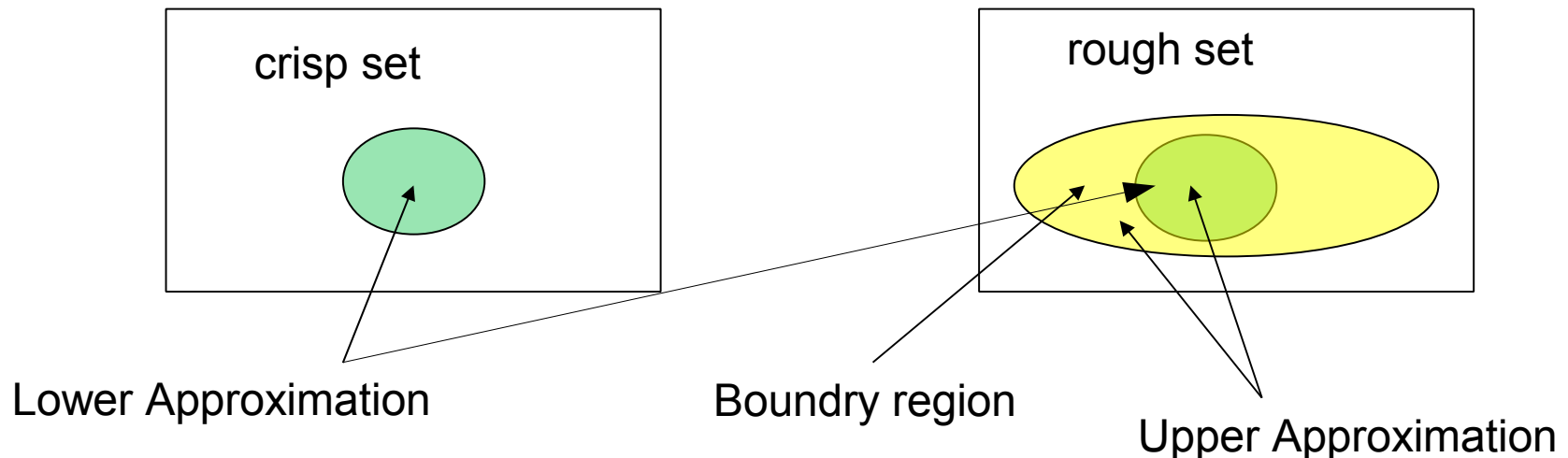
definition Range

d : [real]

# The Rough Sets Theory basics

## Rough set [2 Page 10]

- Extension to a crisp set, where all examples clearly are assigned to one set
- Has lower approximations, examples are clearly classified
- Has upper approximations, where example maybe belong to a class
- Boundry region, between upper and lower approximations
- Boundry region, some examples are not decidable







# The Rough Sets Theory basics

---

## I the indiscernity relation

**reflexive, symmetric, transitive**

Builds up equivalence classes, where objects are indiscernible.

Easier, it works like multiple „=“, grouping objects by equality of their condition-attributes.

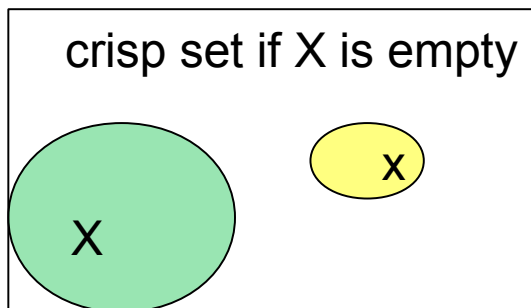
# The Rough Sets Theory basics

## Rough-Membership-Function $\mu$ [2 Page 10]

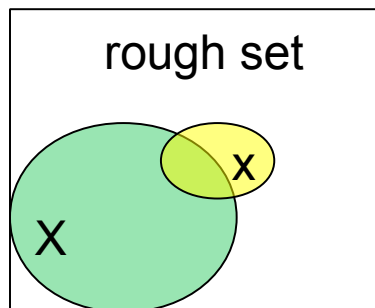
› Gives a probability for an object being member of a set

› Definition  $\mu_X^R : U \rightarrow \langle 0, 1 \rangle$

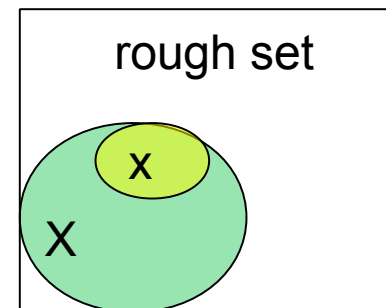
› Function  $\mu_X^R(x) = \frac{|X \cap R(x)|}{|R(x)|}$



$$\mu(x) = 0$$



$$0 < \mu(x) < 1$$



$$\mu(x) = 1$$



# The Rough Sets Theory basics

## Generating Boundry, Upper- and Lower-Approximation

### needed things:

- › take a set of condition-attributes
- › choose a decision-attribute
- › I indiscernity relation
- › choose a desired decision-value
- › a set of examples

### todo:

- › Use „I“ with the decision attribute → sets of equivalent objects for decision: D
- › Use „I“ with the condition attributes → sets of equivalent objects: C
- › Take the set of chosen desired decision value:  $A \subseteq D$
- › Upper Approx. Pup: Find all elements of A in C. If found one, then take the whole equivalence class into Pup.
- › Lower Approx. Plo: like Pup, but takes only objects if whole equivalence class  $\subseteq A$
- › Boundry :  $B_n = \text{Pup} / \text{Plo}$





# The Rough Sets Theory basics

---

## Reducts

- Superfluous condition-attributes, which can be removed without worsen the result
- Reducts are reduced condition-sets
- Often there are several reducts
- Superfluous c-attributes are found during generating Plo and Pup
- Without heuristics it is NP hard problem finding the best removal candidates [1]
- A property of superfluous attributes: their values are coupled to attribute's value

## Benefits

- Less parameters makes classification cheaper
- The Plo will be larger, because less conflicts

# The Rough Sets Theory basics

## Example needed?

Object	Age	BMI	Fat %	Leptin Level
x1	young	good	low	low
x2	middle	medium	low	high
x3	middle	medium	low	low
x4	old	medium	low	high
x5	middle	good	high	high
x6	young	medium	high	low



# Automated rule extraction technique without previous reduct computation – short (5)

$q_i \in Q$ :  $Q$  is finite set of condition – attributes

$f(x, q)$ : is the information function, it delivers the value of a  $q$  object  $x$

$R(x, y)$ : Equality relation

$v_i \in V_q$ :  $V_q$  is a set values of condition – attribute  $q$

The heart of (5)

$$C_{i,j} = \{x : x \in U, R(f(x, q_i), v_{i,j}) = \text{true}\}$$

This operation compares all objects  
with  
all their attributes  
**pairwise**  
to build up equivalence classes



# Automated rule extraction technique without previous reduct computation – short (5)



(5) builds up a tree with  $C_{i,j} = x : x \in U, R(f(x, q_i), v_{i,j}) = \text{true}$

## Properties:

- every rule has this form  $a \rightarrow b$
- a antecedent part
- b consequent part
- that means rules always lead to a decision for desired value in decision attribute
- each stage of the tree produces a rule
- that is described as a manner of DFS in the paper





# Automated rule extraction technique without previous reduct computation – short (5)

## Example

Object	Age	BMI	Fat %	Leptin Level
x1	young	good	low	low
x2	middle	medium	low	high
x3	middle	medium	low	low
x4	old	medium	low	high
x5	middle	good	high	high
x6	young	medium	high	low

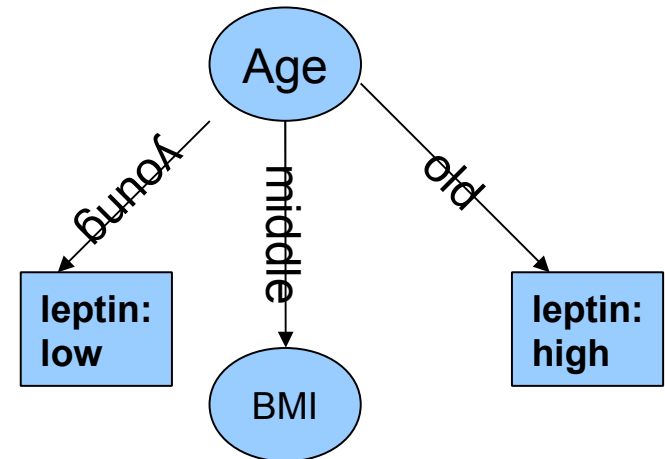
**Target:**  
**finding rules for**  
**low Leptin-level**



# Automated rule extraction technique without previous reduct computation – short (5)

## Example

Object	Age	BMI	Fat %	Leptin Level
x1	young	good	low	low
x2	middle	medium	low	high
x3	middle	medium	low	low
x4	old	medium	low	high
x5	middle	good	high	high
x6	young	medium	high	low



Rule1: Age=young → leptin=low  
Rule2: Age=old → leptin=high

# Automated rule extraction technique without previous reduct computation – short (5)

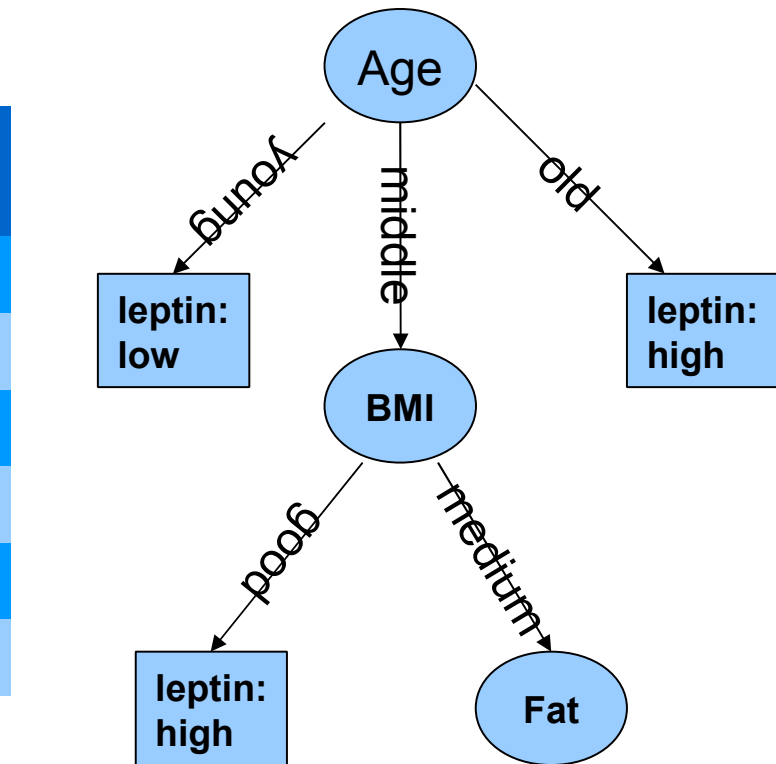
## Example

Object	Age	BMI	Fat %	Leptin Level
x1	young	good	low	low
x2	middle	medium	low	high
x3	middle	medium	low	low
x4	old	medium	low	high
x5	middle	good	high	high
x6	young	medium	high	low

Rule1: Age=young

Rule2: Age=old

Rule3: Age=middle AND BMI = good



→leptin=low

→leptin=high

→leptin=high

# Automated rule extraction technique without previous reduct computation – short (5)

## Example

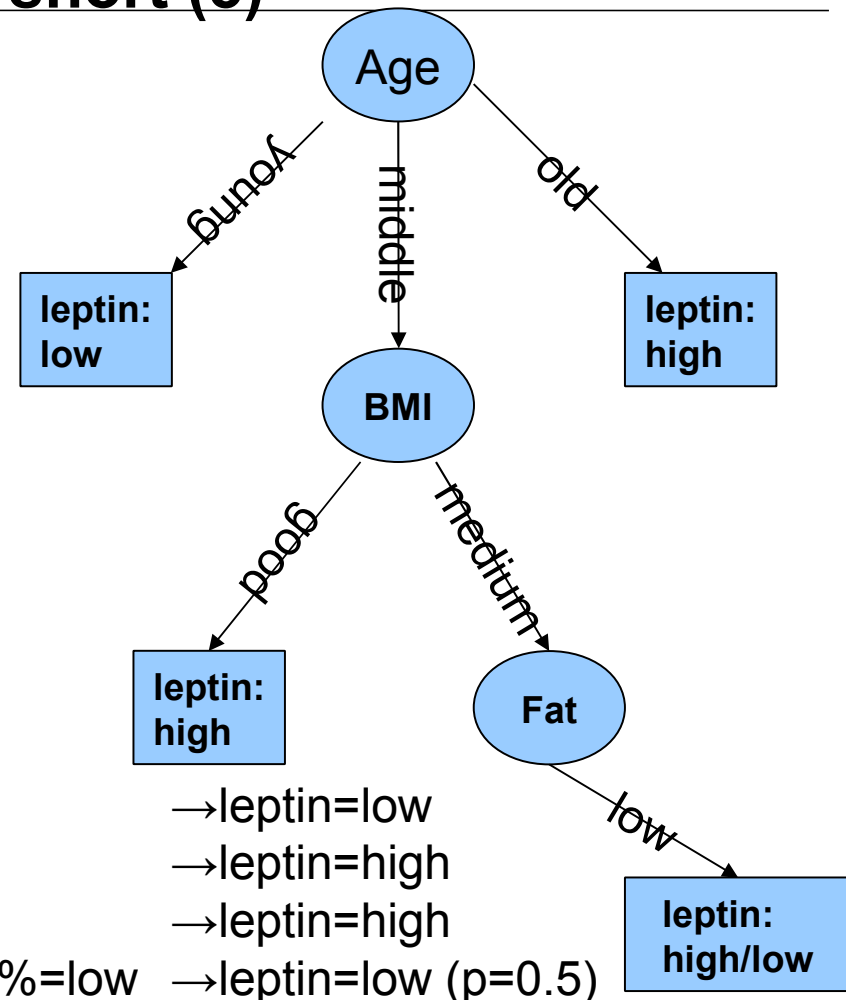
Object	Age	BMI	Fat %	Leptin Level
x1	young	good	low	low
x2	middle	medium	low	high
x3	middle	medium	low	low
x4	old	medium	low	high
x5	middle	good	high	high
x6	young	medium	high	low

Rule1: Age=young

Rule2: Age=old

Rule3: Age=middle AND BMI = good

Rule4: Age=middle AND BMI = good AND Fat%=low → leptin=low (p=0.5)





# Automated rule extraction technique without previous reduct computation – short (5)

## Example

### What happened?

- Rules 1,2 and 3 were extracted for lower approx.
- All Rules build up the upper approx.
- The rules are as short as possible
- The x2 and x3 have same condition-attribute-values
- Age is needed
- BMI and Fat% values are coupled, so they can be mutually changed
- The result is the same as the original rough set approach
- **There was no reduction step necessary!**

# Automated rule extraction technique without previous reduct computation – short (5)



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

**The idea to create (5) this way came from a theorem prover LogPro**

- LogPro works with ordered linear resolution
- LogPro works better than Prolog, which is based on linear resolution with selection function for definite clauses and negation as failure
- No further information [10] [11] papers not found





## Used data for comparison

---

### A medical database from Leptin research

- 36 parameters including Leptin
- Leptin is produced in the fat-cells
- Leptin lets you feel replete
- Leptin is involved in the regulation of food intake
- There are Leptin resistant people
- Leptin resistance is often associated with insulin resistance
- Database has been already used for another paper[12], I did not find  
I believe, the ROSETTA program, they used for this paper,  
is described there!



## Discretization for real values

---

Discretization is an issue of importance without that rough set theory would not work well.

### **ROSETTAs discretization [1 Pages 15-20]**

1. sort the values of attributes and reducing the values to a set
2. calculate average-values between every attribute value. These intervall cutpoints are assigned to propositional variables  $p$ .
3. cluster the real values. This is NP hard, but there are good heuristics.

## Discretization for real values

Rosetta uses MD-Heuristics like described in [5],  
I hope I understood that reference right, it is written strangely

### Indiscernibility matrix for MD-Heuristics

1. build up a discernibility matrix in the following way:

Example condition-attributes A, B  
with Instance a1, a2, a3, b1, b2  
for rows.

The columns carry cut off points

2. fill it up with: 1 for every entry where  
 $\min(x,y) < p < \max(x,y)$   
and the rest is 0.

A	p1a	p2a	p3vb	d*
(a1, b1)	1	1	0	1
(a1, b2)	1	0	0	1
(a2, b1)	0	1	1	1
(a2, b2)	0	1	0	1
(a3, b1)	0	0	1	1
(a3, b2)	0	0	0	1
new	0	0	0	0



# Discretization for real values

## MD Heuristics.

- „1. Construct the Indiscernibility matrix as described and delete the last row with the new element. set  $B:=A$
2. Choose a column from B with maximal occurrences of 1's
3. Delete from B the column chosen in Step 2 and all rows marked in this column by 1
4. If B is non-empty then go to Step 2 else stop.“

Ac	p1a	p2a	p3b	d*
(a1, b1)	1	1	0	1
(a1, b2)	1	0	1	1
(a2, b1)	0	1	1	1
(a2, b2)	0	1	0	1
(a3, b1)	0	0	1	1
(a3, b2)	0	0	0	1
new	0	0	0	0

p2a: first cut off point

## Discretization for real values

A	p1a	p3b	d*
(a1, b2)	1	1	1
(a3, b1)	0	1	1
(a3, b2)	0	0	1

Cut off points:  $p3vb$ ,  $p2a$

A	p1a	d*
(a3, b2)	0	1

Cut off points:  $p3vb$ ,  $p2a$

The cut off points divide planes.

They construct lines orthogonal to the axis of a attribute in a graph. The equivalence classes are between the lines.

## Discretization for real values

---

The editors wrote, that only 8 parameters were discretized, because the others were omitted.

I cannot prove that, I just have to belief it.

It sounds, that their programm (5) had the same data as ROSETTA.

They let Matlab classificate the Leptin values in two classes with standardized euclidean distance and centroid distance.

No further information.

## Results of comparison ROSETTA and (5)

### Results:

- Both generated 70 rules
- ROSETTA's Rules have more literals, always 8
- (5)'s are direct pruned, as prognosed in (5)'s description

**Table 3. If - Then rules generated by ROSETTA**

IF Age([32, *]) AND Waist([*, 101]) AND HDL([1.05, *]) AND nonHDL([4.03, *]) AND FSI([*, 8.95]) AND FFM(kg)([*, 55.1]) AND TFat%([*, 44.8]) THEN Leptin[*,24.925)
IF Age([30, 32]) AND Waist([*, 101]) AND HDL([1.05, *]) AND nonHDL([*, 4.03]) AND FSI([*, 8.95]) AND FFM(kg)([*, 55.1]) AND TFat%([*, 44.8]) THEN Leptin[*,24.925)
IF Age([*, 30]) AND Waist([108, 114]) AND HDL([*, 1.05]) AND nonHDL([4.03, *]) AND FSI([8.95, *]) AND FFM(kg)([55.1, *]) AND TFat%([44.8, *]) THEN Leptin[24.925,*)
IF Age([*, 30]) AND Waist([*, 101]) AND HDL([*, 1.05]) AND nonHDL([4.03, *]) AND FSI([*, 8.95]) AND FFM(kg)([*, 55.1]) AND TFat%([*, 44.8]) THEN Leptin[*,24.925)

**Table 4. If - Then rules generated by technique based on (5)**

IF Age[*,30) AND Waist[*,101) AND HDL[*,1.05) AND nonHDL[*,4.03) AND FSI[*,8.95) THEN Leptin[24.925,*)
IF Age[*,30) AND Waist[*,101) AND HDL[*,1.05) AND nonHDL[4.03,*) AND FSI[*,8.95) THEN Leptin[*,24.925)
IF Age[30,32) AND Waist[*,101) AND HDL[1.05,*) THEN Leptin[*,24.925)
IF Age[30,32) AND Waist[101,108) THEN Leptin[24.925,*)



## Conclusion and remarks

---

### **and operators**

Rosetta: 420

With 70 rules: every rule had all 7 condition-attributes

(5): 297

**The main target of this paper is achieved**

**A method is found to produce easier inspectable rules**

**(5)'s rules are more concise than those of ROSETTA**

### **Other benefits**

(5)'s rules are cheaper to use in classification

(5) saves a NP-hard reduction of attributes



## Future outline

---

### **The editor want to:**

- Prove how good the generalization of the learned rules will be with new medical data.
- The editors want to improve generalization, so that arbitrary data is classified well.



# Critique

---

## Testdata

- Only one test database was used and no other
- No Reference is given, where the data is found
- No sample of test-data available to look over
- They are right to do further tests.

## Experiment

- Only one competitor program
- Accuracy and Precision?

## Paper itself

- Not easy to find citations in the references
- Sometimes explanation too short
- Many references, I did not find!

# In the paper Data-Driven Adaptive Selection of Rules Quality Measures for Improving the Rules Induction Algorithm



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

## Abstract

- **Comparing different measure methods**
- **Showing the influence of „filtration“ on them**
- **Presenting a automatic measure selector**
- **Experiment with 21 benchmark data sets**





# Introduction

---

## **Inductive rule learner are popular**

- their results are patterns in rule form
- their algorithm structure is clear
- rules are simple
- rules are explainable

## **The editors want to research how inductive learners can be improved**

- chosen way: learn about measure methods
- measure methods steering, which rules add and pruned in rule sets
- combine measure methods with filtration
- testing an automatic measure method selector
- their work is inspired by An's paper [16]

# Introduction

## An's Paper [16] „Rule Quality Measures for Rule Induction Systems: Description and Evaluation“

- Rule Induction Systems and quality measures for rules are examined
- Learning system ELEM2, written by editors of An's paper are used to test the measures
- ELEM2: sequential covering, which leads to a small overfitting, pruning controlled by a measure method
- Used measures: MD, Cohen, C1, C2, IS, LS, WS, Prod,  $G^2$  likelihood and  $\chi^2$
- ELEM2 meta learner variant is tested: learns rules for measure selection
- ten-fold tests on 27 datasets
- their data sets came from UCI Repository of Machine Learning database

This paper is strictly orientated on An's paper, even the structure is similar, but **this paper is not as detailed as An's paper[16]**.



# Introduction

---

Continuation of:

Decision Rule-Based Data Models Using TRS and NetTRS - Methods and Algorithms[17]

Rule quality measures in creation and reduction of data rule models[18]

[17] and [18] not downloaded

Influenced by:

On the Quest for Optimal Rule Learning Heuristics[19]  
not downloadable

# Introduction

## A Typical Induction Learner

popular examples: RIPPER, CN2, AQ family and rough set theory learner

- covering at least all positive examples
- if only positive examples covered, then it overfits
- the rules can be stronger (covering more examples) or weaker
- Sometimes weak rules are completely covered
- In general the learners have 3 phases
  - Growing: Rule induction, building a rule set with aid of measures
  - Pruning: rule set reduction, until precision decreases with measures aid
  - Filtering: rule set pruning, until accuracy falls with constant precision



# Introduction

---

## **A good Learner needs good measure methods**

- overfitted rule sets only recognize during classification, what they have seen in training
- rule sets have to be general enough to recognize also similar objects like examples of training-data
- good rule sets are as small as possible
- covered weak rules have to be scored for necessity
- not every measure method fits to all data sets

finding high quality rule sets requires:  
Measure Methods

# Measure Methods

## PN space [20]

p	n	p+n
P-p	N-n	P+N-p-n
<b>P</b>	<b>N</b>	<b>P+N</b>

## Basic formels

$$precision = \frac{p}{p+n}$$

$$accuracy = \frac{p+N-n}{P+N}$$

$$coverage = \frac{p}{P} \quad \text{also called recall}$$

- p: count of positive examples as positive recognized
- n: count of negative examples as negative recognized
- P: all positive examples
- N: all negative examples

## The measure methods examined in this paper

laplace[20] : á priori probality 50%

$$h_{lap} = \frac{p+1}{p+n+2}$$

*p*: count of positive examples as positive recognized  
*n*: count of negative examples as negative recognized

m-estimate[20]: á priori probality adjustable

$$h_m = \frac{p+m*P/(P+N)}{p+n+m}$$

**m=0** ~ precision

**m>0** ~

**m<0** ~ **ditto**  $h_{lap}$

*p*: count of positive examples as positive recognized  
*n*: count of negative examples as negative recognized  
*P*: all positive examples  
*N*: all negative examples  
*m*: prior total coverage in the training set



# Measure Methods

**g - measure[20]: precision adjustable**

**g=0 ~ precision**

**g>0 ~ more precise → overfitting**

**g<0 ~ less precise → generalizing**

$$h_{lap} = \frac{p}{p+n+g}$$

*p: count of positive examples as positive recognized*

*n: count of negative examples as negative recognized*

*g: adjusts precision formula*

**WRA measure[20]: equivalent of weighted relative accuracy**

**Proof in [20 page 10]**

**Only accuracy based is not really mighty.**

$$h_{WRA} = \frac{p}{P} - \frac{n}{N}$$





# Measure Methods

TABLE 2. Contingency Table with Relative Frequencies

	Class $C$	Not class $C$	
Covered by rule $R$	$f_{rC}$	$f_{r\bar{C}}$	$f_r$
Not covered by $R$	$f_{\bar{r}C}$	$f_{\bar{r}\bar{C}}$	$f_{\bar{r}}$
	$f_C$	$f_{\bar{C}}$	1

Frequencies

coresspond to the NP space table.

All entries of NP space are divided by  $P+N$ , thats table with Relative Frequencies.

**Cohen[16]: relative accuracy based**  
measure the indepence, with a kind of regression.  
Cohen is nearer 1, when for raising  $p$  and  $N-n$  in  $P+N$ .

$$Cohen = \frac{f_{rC} + f_{\bar{r}\bar{C}} - (f_r * f_{\bar{C}} + f_{\bar{r}} * f_C)}{1 - (f_r * f_{\bar{C}} + f_{\bar{r}} * f_C)}$$



# Measure Methods

## C2 [16] : modified Coleman's Formula

### Coleman's Formula:

measures modifies precision in way,  
that rating goes nearer 1, when less positive  
examples in training data.

$$Coleman = \frac{precision(R) - f_c}{1 - f_c}$$

### C2:

measure is improved it supports high  
precision and high recall.

This should recognize everything, what works  
good!

$$C2 = Coleman \frac{1 + recall(R)}{2}$$





# Measure Methods

---

**p-val [19]: Paper was not downloadable :(**

**There was tried to learn stastical about rules.**

**Rules were described with 9 parameters.**

**A SVN learned with rule sets of 30 benchmark data-sets.**

**The SVN created pval and nval from 9 parameters of each rule to find out, which conditions leads to good rules.**

**Did I understand this right Dr. Jansen?**



# Measure Methods

---

**LIFT[23 Page 4]:**

**Is a measure Method, which is comparing a random choice with prediction probability of the rule set. The measure is only the ratio of both probabilities. The counter has the probability of the rule set.**



# Measure Methods

---

**Measure Method of RIPPER [4]:**

**in the RIPPER part of presentation:**

**RIPPER is the competitive program to the q-MODLEM, which the editors use to test.**



# Measure Methods

---

**Gain:** is described in [17].  
I could not download that paper.

**RSS and C2F:** I have not seen in a Reference, I got in my fingers. Searching for them was not satisfying.



# Filtration

**All measure methods are combined with filtration.**

## **What is filtration?**

- A step after growing and pruning a rule set
- Prunes rule which can be thrown out of a rule set without decreasing Precision and Accuracy
- It follows Occam's razor

The papers shows how good the measures work  
**with** the three filtration approaches  
and  
**without** filtration.





# Filtration

## Filtration Methods:

- **Forward filtration:** Packs in an empty rule set, the learned rules until accuracy does not grow anymore
- **Backward filtration:** Removes rules until accuracy decreases
- **Coverage filtration:** Build up a new rule set. Rules are measured and ranked by a measure method. It chooses the best rules until all examples are covered. It remove lower quality rules.





# The induction algorithms of this Experiment

---

## **q-MODLEM Algorithm:**

- It based on MODLEM
- Used in another paper [18] for testing measure methods

## **q-MODLEM with automated measure method selector:**

- Starts a crossvalidation for all measures methods
- Chooses the best measure for each database

## **Ripper program:**

- Inductive Rule learner
- Well known, I suppose
- Used unmodified in the experiments
- Build in rule pruner

# MODLEM

## MODLEM [22 Page 2]

- Inductive rule learner
- Produces rules from type: IF (condition-attributes) then decision equivalence class similar to slide 7
- Main idea: sequential covering of examples with Rough Sets Theory
- Heuristics help to produce minimal rule sets
- Examples covered by rules are removed
- If there are only examples left with inconsistent descriptors then MODLEM uses
  - either Laplace measure
  - or Class Entropy measure for generating the uncertain rules.



# MODLEM

## Features[22]

- Uses direct real values in condition attributes without discretization of condition attributes
  - Numerical values are represented by
$$a < v_a \quad \text{or} \quad a \geq v_a$$
$$v_a$$
 is a threshold to value to attribute class A, calculated with a measure.
- One condition-attribute can be used several times in a rule; leads to intervals



## q-MODLEM

### Differences between MODLEM and q-MODLEM

- In q-MODLEM measure methods are exchangeable
- Has now Climbing Hill Strategy: Growing and Pruning
- Growing phase until no rule are found increasing precision
- Pruning phase removes elementary conditions (literals?) from rules as long as precision keeps.
- Optional filtration step with described filtration methods.



# RIPPER Program[21]

**RIPPER Program is combination of combined steps with growing and pruning and has its root in REP → IREP → IREP\* → RIPPER.**

➤ In each step:

- training data is separated randomly to growing and pruning set
- induction of a new rule with overfitting
  - Induction stops either no example is left
  - or new are longer than a chosen delta of bits between new rule and longest existing rule
- pruning the rules literals as follows: 3 versions of the new rule:
  - Variant A like IREP\*: pruning with a precision like measure:  $measure = \frac{p+N-n}{P+N}$
  - Pruning Variant A towards minimal error
  - Adding literals from other rules to Variant A and measure it
- MDL Heuristic decides which version is the best rule.

The measure has been proved as precision [20 Page 12]





# The Classifier

---

## The classifier

- no references
- no name
- no idea of quality
- description
  - using a voting scheme
  - the quality of a rule is its voting strength
  - not decided examples are assigned to majority decision class

# Experiments

## The experiment is really easy

**Ripper, q-MODLEM with all measure and filtration combination and q-MODLEM with automated measure selector.**

- Have to work through 21 Benchmark datasets
- Ten fold cross-validation
  - testing measure methods
  - selection of the best measure for a data set
- 85% of a data for cross-validation
- 15% for tuning during filtration
- Results expressed by
  - Accuracy =  $(p + N - n) / (P + N)$
  - AVG Accuracy
  - Count of Rules
  - Conflicting Examples %
  - Conflicting Examples % solved false
  - p-Value of Wilcoxon test,

### Benchmark Data Sets

balancescale, breast-wisc, bupa, car, Australian credit, German credit, diabetes, ecoli, glass, heart Cleveland, heart statlog, ionosphere, iris, kdd-synthetic-control, lymphography, prnn-synthetic, segment, sonar, splice, wine, yeast

# Results

Table 1. Average results of 10xCV experiments

Quality	Filtration	Acc.	Std.	Avg.	Std.	Rules	Std.	Confl.	Neg.	p-Value
				Acc.					Confl.	
Auto	None	83.51	5.2	77.61	6.4	144.4	24.2	33	8	-
Auto	Coverage	83.09	5.0	76.36	6.1	50.5	10.0	30	8	0.03809
g-measure	None	82.86	5.6	75.62	6.3	169.4	7.8	27	8	0.08881
C2	None	82.56	5.1	76.09	6.1	148.4	8.1	28	8	0.06525
C2F	None	82.45	4.5	74.50	5.4	135.1	7.2	36	11	0.12862
m-estimate	Coverage	82.33	5.0	74.53	5.8	58.5	4.4	42	13	0.00587
Lift	None	82.23	5.7	79.43	6.7	185.9	7.3	24	9	0.05553
C2	Coverage	82.21	4.6	75.41	5.5	43.1	3.2	17	5	0.01913
m-estimate	None	82.18	5.0	74.35	5.7	101.7	7.6	45	13	0.00030
C2F	Coverage	82.06	4.7	73.89	5.6	52.3	3.6	27	8	0.06864
g-measure	Coverage	82.01	5.2	74.68	6.1	62.7	4.2	17	5	0.00380
Auto	Backwards	81.84	5.3	75.08	6.3	21.5	6.4	33	9	0.00059
m-estimate	Forward	81.80	5.0	75.40	6.1	29.1	6.9	21	7	0.00036
m-estimate	Backwards	81.68	5.5	75.32	6.5	27.1	6.7	25	7	0.00093
Auto	Forward	81.52	5.2	75.53	6.3	21.3	8.6	26	8	0.00018
Lift	Coverage	81.40	5.5	76.80	6.4	54.9	3.5	11	4	0.00277
Cohen	Backwards	81.18	6.1	75.09	6.8	17.3	4.4	48	14	0.00036
RIPPER	-	81.09	5.02	74.27	6.24	18.9	3.4	-	-	0.00277
C2	Forward	80.97	4.6	74.11	6.2	23.6	6.4	9	2	0.00012
C2F	Forward	80.90	4.9	73.19	6.0	25.5	6.7	14	4	0.00036
Cohen	Forward	80.86	5.7	76.14	6.6	10.7	2.4	36	11	0.00016
p-Val	Backwards	80.86	5.8	73.67	6.5	13.6	3.7	46	13	0.00028
Cohen	None	80.83	5.8	76.34	6.3	40.4	4.0	66	19	0.00093

Quality	Filtration	Acc.	Std.	Avg.	Std.	Rules	Std.	Confl.	Neg.	p-Value
				Acc.					Confl.	
Cohen	None	80.83	5.8	76.34	6.3	40.4	4.0	66	19	0.00093
C2	Backwards	80.81	4.9	74.42	5.9	20.2	5.2	9	2	0.00005
Gain	Forward	80.80	5.0	75.34	6.3	10.4	2.5	41	12	0.00003
Gain	Backwards	80.80	5.6	73.74	6.5	12.8	3.3	47	13	0.00001
C2F	Backwards	80.79	5.1	73.69	6.1	21.6	5.7	16	4	0.00028
Cohen	Coverage	80.71	5.8	76.21	6.4	26.0	2.6	65	19	0.00059
Wra	Backwards	80.43	5.2	73.83	5.7	14.6	3.7	57	16	0.00004
p-Val	Forward	80.41	5.7	74.26	6.5	10.7	3.0	38	11	0.00002
g-measure	Forward	80.20	5.1	73.27	6.7	33.8	7.2	7	2	0.00003
g-measure	Backwards	80.07	5.3	73.62	6.6	28.9	5.5	8	2	0.00004
Rss	Backwards	79.98	5.4	73.36	5.9	14.6	3.5	57	16	0.00002
Lift	Forward	79.79	5.2	74.02	6.3	31.6	7.7	5	2	0.00002
p-Val	None	79.66	5.7	73.00	6.4	39.0	4.8	70	23	0.00119
p-Val	Coverage	79.59	5.6	73.20	6.3	24.5	2.8	68	23	0.00032
Lift	Backwards	79.47	5.6	73.64	6.6	27.7	6.4	5	2	0.00003
Gain	None	79.35	5.2	72.92	5.9	34.6	5.0	71	23	0.00054
Wra	Forward	79.20	5.3	74.96	6.0	8.6	2.1	48	16	0.00001
Rss	Forward	79.18	5.3	74.88	6.0	8.6	2.0	47	16	0.00010
Gain	Coverage	79.16	5.3	73.00	6.3	22.8	2.9	70	23	0.00003
Rss	Coverage	76.58	5.7	71.79	6.4	19.3	2.3	79	28	0.00010
Wra	Coverage	76.46	5.7	71.68	6.4	19.4	2.2	79	28	0.00000
Rss	None	75.66	5.4	70.68	6.5	28.3	3.5	80	31	0.00018
Wra	None	75.61	5.5	70.62	6.5	28.3	3.5	81	31	0.00016

m from m-estimate = 22,4  
g from g-measure = 2

No explanation for m, g  
is given.



# Results

- The result is speaking for the automatic selection
  - I do not understand what mean with:

„It’s worth noticing that improving classification accuracy in the automatic method doesn’t happen at expense of average accuracy loss(which affects sensitivity and specificity of the classifier)“

- choosing the best measure should lead to best result
- There is a trend: larger rule have better results
  - this leads to critique on Occam’s Razor

# Results

- Filtration seems only to help for weak measure methods
- Rule sets with less rules seems to have less conflicts
- Filtration worsend the results of the measures
  - with exception of m-estimate that lost 43 rules and was 0.15% more accurate

## My remarks

- the weaker measure are often accuracy based and the better one on precision
- filtration works on accuracy basis in this experiment
  - perhapps accuracy based can win, because they're their rules have no preference for positive and negative examples (Winner: Cohen, WRA and ? )
  - precision based usually loses, because their rules preferring positive examples. Losers( C2, C2F, g-measure and ?) Winner(m-estimate)m

## Conclusions and future outline

---

Automatic selection is better than arbitrary

- This time it was optimized for accuracy
  - Next time it could be optimized for AVG accuracy
- OR
- try complex measures based on accuracy and model complexity

The filtration reduces rules at cost worsening the accuracy

- In future the filtration could be improved

The classifiers voting algorithm could be improved by

- taking the neighborhood into consideration
- using statistical significance of rules

# Critique

The filtration was not so bad as described if yo take a look on accuracy it worsend not much.

- the benefits for classificator which have to work on low capacity enviroments are ignored

q-Modlem and RIPPER were the only programs tested during experiments

- more competing programs, would be enlighting
  - we do not know, if another learner would total different results
- dito for another classifier
- it this right to have RIPPER in the experiment
- a comparison with ELEM2 from An's[16] paper could have been done

The paper was writte quite well, large parts were understandable without references, but not the measure methods.

Papers like An's[16] are better to read.

# Critique

---

**The main aims are achieved**

- **The automated measure selector, worked well**
- **Not every measures method fits to every data set**

# Similarities between the papers and common future outline



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

## Shortcuts for papers titles:

- [A] A Comparison of Rule Sets Generated from Databases by Indisceribility Relation – A Rough Sets ApproachIn the paper Data
- [B] Driven Adaptive Selection of Rules Quality Measures for Improving the Rules Induction Algorithm

Both papers approaches

- are to prune rule sets and keeping precision and accuracy
- are driven from readability of rule sets for experts
- have automated the pruning
- worked with Rough Set algorithm



# Similarites

---

## Both papers approaches

- are to prune rule sets and keeping precision and accuracy
- are driven from readability of rule sets for experts
- have automated the pruning
- worked with Rough Set algorithm
- were successful



## Common remarks

---

### Differences between the papers

- different main aims
  - [A] Rule Set Quality: count of rules and count of literals
  - [B] Accuracy, count of literals, Conflicts during classification
- [A] and [B] are not comparable
  - In [A] is a data set used diabetes, if its parameter are largely equal to the medical data of [B] then you could do a comparison with look on rule count
  - The paper's results does not fit together, but the used programm would fit in a common experiment.





## Common remarks

---

### Common future outlines

- the programs of [A] and [B] could be compared, that match to [A]'s and [B]'s outlines to further tests with new data
- you could try to create a new inductive learner which uses the ideas of (5) and q-MODLEM with automated measure selection
  - its feature:
    - working direct on real values
    - chooses the best measure method
    - decision tree with minimal rule length without reducts
-



# References

- [1] Modelling prognostic power of cardiac tests using rough sets  
<http://www.ncbi.nlm.nih.gov/pubmed/10082180>
- [2] Rough Sets; Zdzisław Pawlak  
<http://bcpw.bg.pw.edu.pl/Content/2026/RoughSetsRep29.pdf>
- [3] **Rough Set Approach to Domain Knowledge Approximation**
- [5] Rough Sets: A Tutorial
- [6] Implementierung und Testen eines relationalen Lern-Algorithmus
- [10]
- [11] The AQ15 Inductive Learning System: An Overview and Experiments
- [15] An Entropy gain Measure of Numeric Prediction Performance
- [16] Rule Quality Measures for Rule Induction Systems: Description and Evaluation
- [17] Decision Rule-Based Data Models Using TRS and NetTRS - Methods and Algorithms
- [18] Rule quality measures in creation and reduction of data rule models
- [19] On the Quest for Optimal Rule Learning Heuristics
- [20] **ROC 'n' Rule Learning—Towards a Better** Understanding of Covering Algorithms
- [21] Fast Effective Rule Induction
- [22] An Experimental Study of Using Rule Induction Algorithm in Combiner Multiple Classifier
- [23] **Discovering Characteristics of Aberrant Driving Behavior**  
<http://www.wseas.us/e-library/conferences/2005corfu/c1/papers/498-484.pdf>

