# Outlier Detection and Explanation with Random Forests

Graph mining and outlier detection meets Logic proof tutoring

Karel Vaculík, Leona Nezvalová, Luboš Popelínský

Knowledge Discovery Lab, FI MU Brno

Seminář strojového učení a modelování, Praha 18/12/14

# Outline

- Resolution for theorem prooving
- Graph data – student solutions
- New features - generalized subgraph patterns
- Class outlier detection
- Finding anomalous graphs
- Explanation of outliers

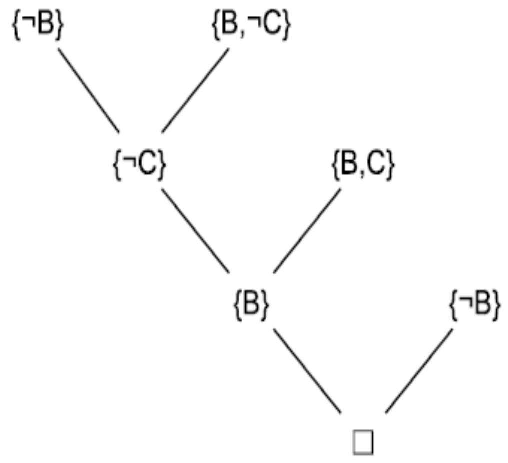
# Introduction

**Resolution in propositional logic**

- Theorem-proving technique based on refutation
- *Resolution rule* produces a new clause implied by two clauses
- *Example*: Prove that the following set of clauses is contradictory

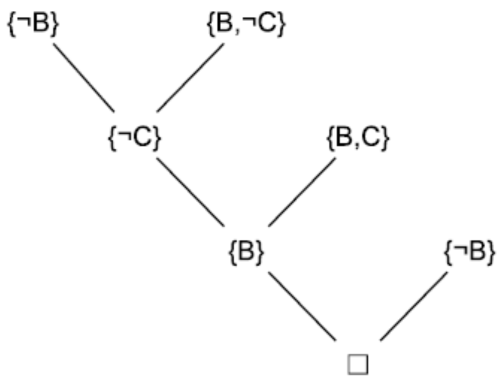
(B or ¬C) and ¬C and (B or C) and ¬B and (¬A,B) and (A or C)

{{B,¬C}, {¬C}, {B,C}, {¬B}, {¬A,B}, {A,C}}



# Method and goal

{{B,¬C}, {¬C}, {B,C}, {¬B), {¬A,B}, {A,C}}

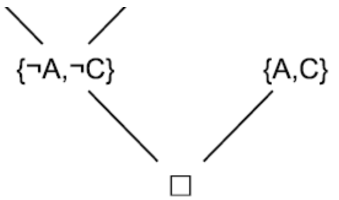



- graph mining, class outlier detection

- teaching enhancement by analysing students' solutions

# Data

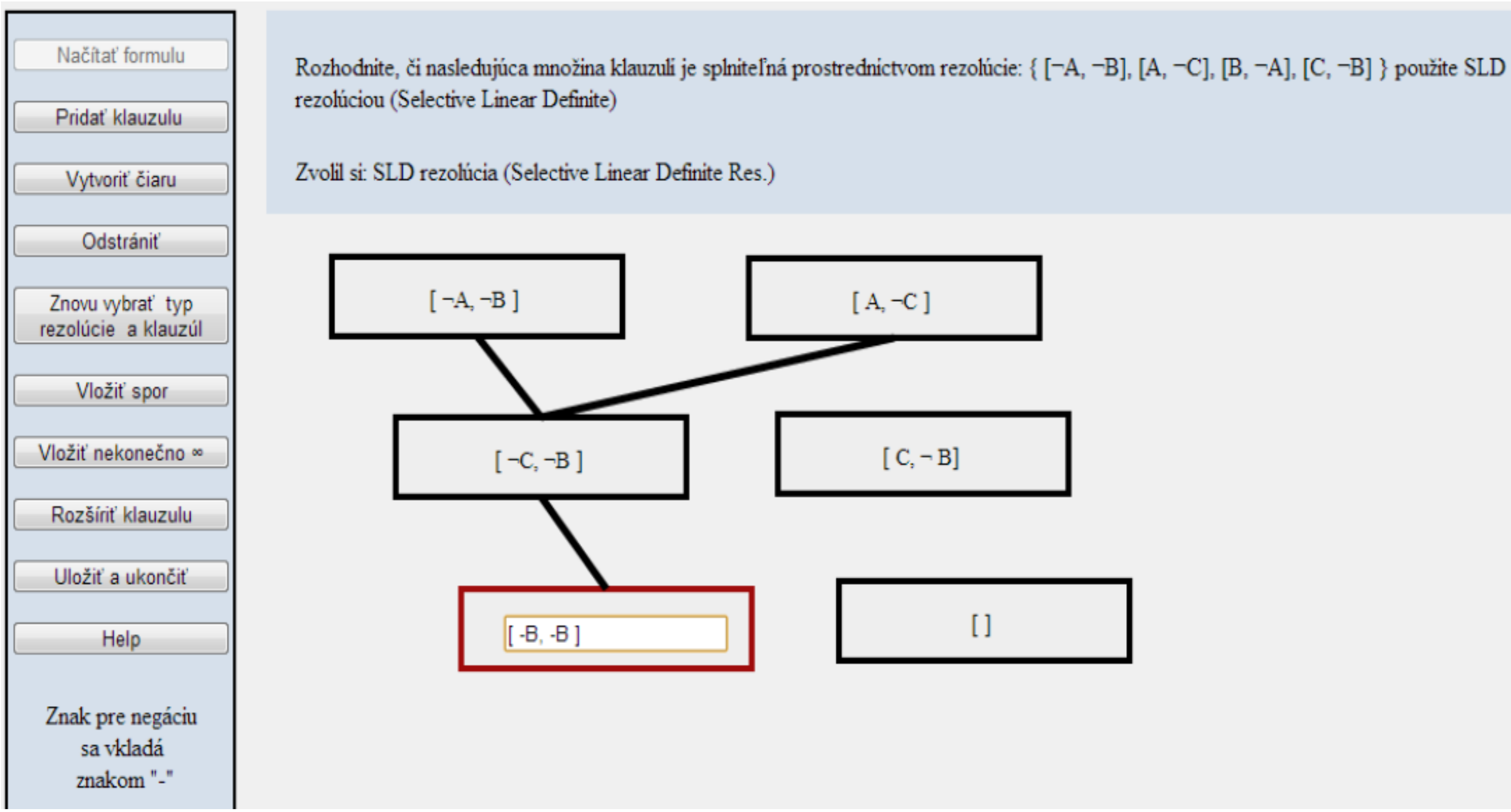
- 351 students solved resolution proofs via a web-based tool




- 873 resolution proofs classified into two classes
  - 772 correct
  - 101 incorrect

Incorrect ~ at least one error has been detected by an automatic evaluator

The most serious and common error: resolving on two literals at the same time


Načítať formulu
Pridať klauzulu
Vytvoriť čiaru
Odstrániť
Znovu vybrať typ rezolúcie a klauzúl
Vložiť spor
Vložiť nekonečno ∞
Rozšíriť klauzulu
Uložiť a ukončiť
Help
Znak pre negáciu sa vkladá znakom "-"
Rozhodnite, či nasledujúca množina klauzuli je splniteľná prostredníctvom rezolúcie: { [¬A, ¬B], [A, ¬C], [B, ¬A], [C, ¬B] } použite SLD rezolúciou (Selective Linear Definite)
Zvolil si: SLD rezolúcia (Selective Linear Definite Res.)
[ ¬A, ¬B ]
[ A, ¬C ]
[ ¬C, ¬B ]
[ C, ¬ B]
[ -B, -B ]
[ ]

# New features : subgraph patterns

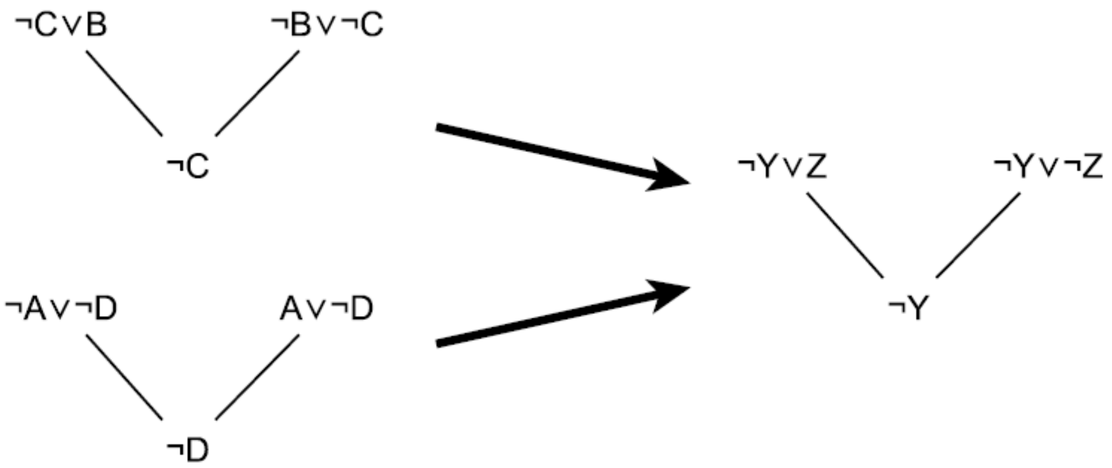
- Representation of graphs by their substructures – patterns
  - *Pattern$_i$* appeared in the graph -> true
  - *Pattern$_i$* did not appear in the graph -> false

| **pattern$_1$** | **pattern$_2$** | **...** | **pattern$_m$** | **class** |
|---|---|---|---|---|
| true | false | ... | false | incorrect |
| ... | ... | ... | ... | |
| false | true | ... | true | correct |

# Generalized Subgraph Patterns

- Representation of graphs by their substructures (patterns)
- Simple subgraphs inappropriate

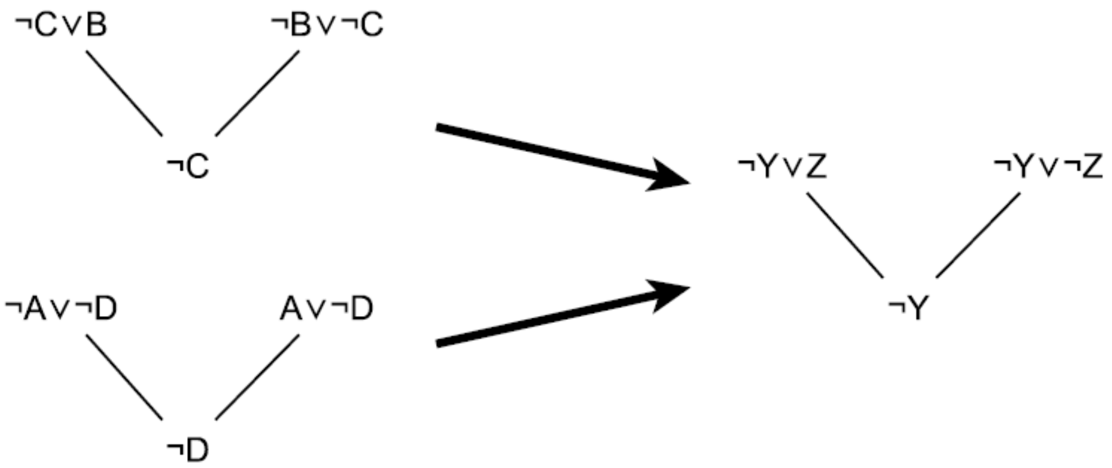
  => generalized subgraphs



# Generalized Subgraph Patterns

Procedure:

1. Extract all 3-node subgraphs (parents with the resolvent)
2. Perform generalization on these subgraphs



# Generalized Subgraph Patterns – Higher Level

- To increase performance of algorithms, we created a new, higher-level, generalization
- This method generalizes patterns created by previous gen. method
- It exploits domain knowledge about general resolution principle
- From 3-node generalized pattern it creates new pattern in following form:

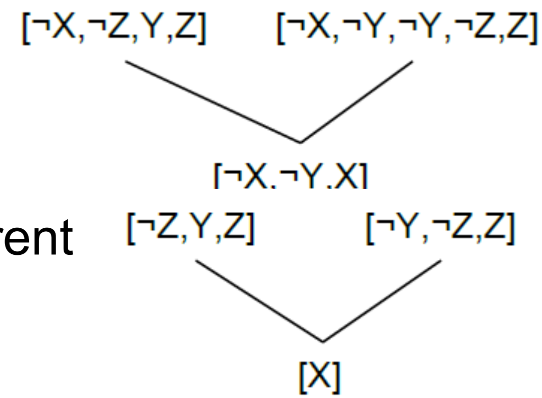
$$[L_{i_1}, L_{i_2}, \dots, L_{i_n}]; [L_{j_1}, L_{j_2}, \dots, L_{j_m}]$$

*(added)* (*dropped)*

- *added*: literals which were added erroneously to the resolvent
- *dropped*: literals from parents which participated in the resolution process

# Generalized Subgraph Patterns – Higher Level

Example:




- Go through literals in resolvent and delete those that occur in at least one parent

*added* = [X], *dropped* = [¬Z,Y,Z,¬Y,¬Z,Z]

- Rename letters in lists *dropped* and *added*, and sort both lists to get the final pattern

[Z];[¬X,¬X,¬Y,X,X,Y]

# Detection of anomalous solutions

(1a) Learn a classifier that would discriminate between correct and incorrect solutions, and

Detect rare cases that are incorrectly classified – outliers

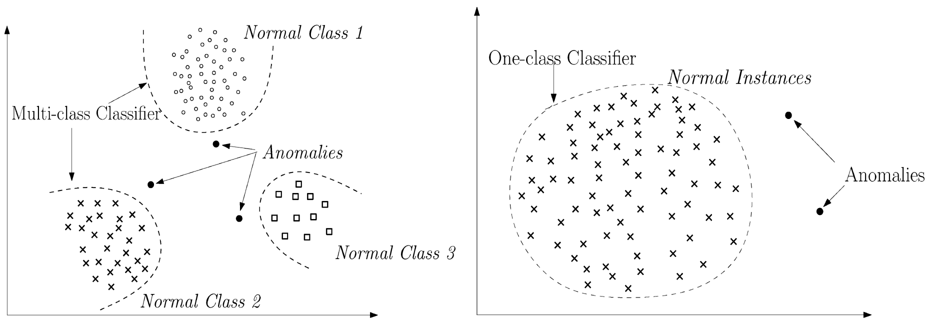
(1b) Clustering and observing class distribution in clusters

(2) BETTER: Directly detect outliers without learning a classifier.

Data have been classified => common outlier detection does not help

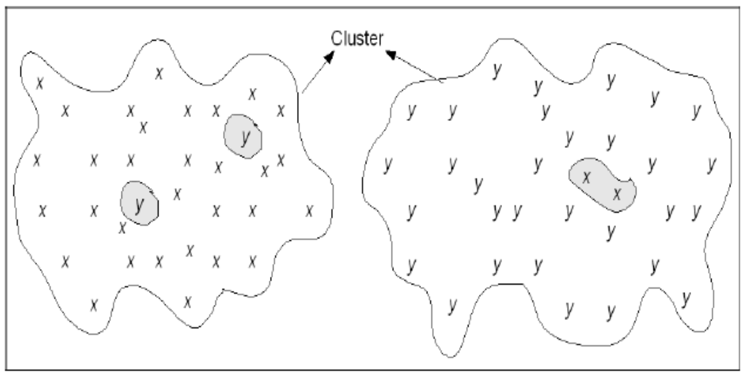
=> Class-based outlier detection

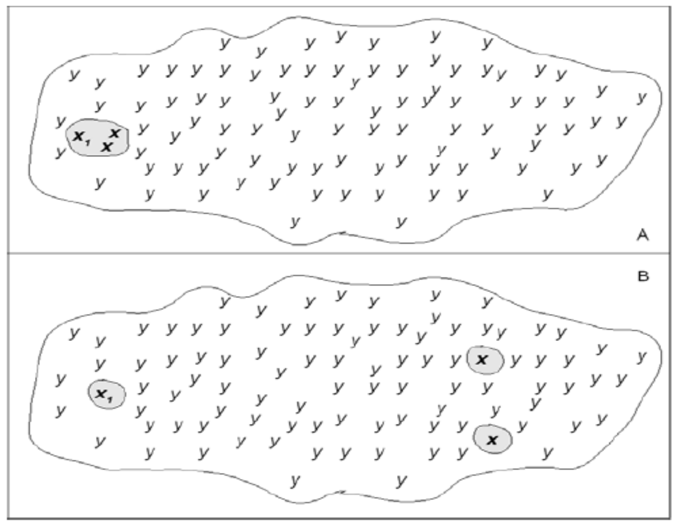
# Class outliers




(a) Multi-class Anomaly Detection

(b) One-class Anomaly Detection



# Class Outlier Detection

CODB (Hewahi and Saad 2007) – distance and density based

- weka-peka  (Pekarčíková 2013)
  - Based on Random Forests
  - Analysis of proximity matrix

# Class Outlier Detection via Random Forest

weka-peka (Pekarčíková 2013)

- Learn Random Forest
- *A*fter each tree is built, all of the data are run down the tree, and proximities are computed for each pair of cases:

  If two cases occupy the same terminal node, their proximity is increased by one.
- At the end of the run, the proximities are normalized by dividing by the number of trees.

# Class Outlier Detection via Random Forest

weka-peka (Pekarčíková 2013)

FO($p$) = FO1($p$) + FO2($p$) + cFO3($p$)

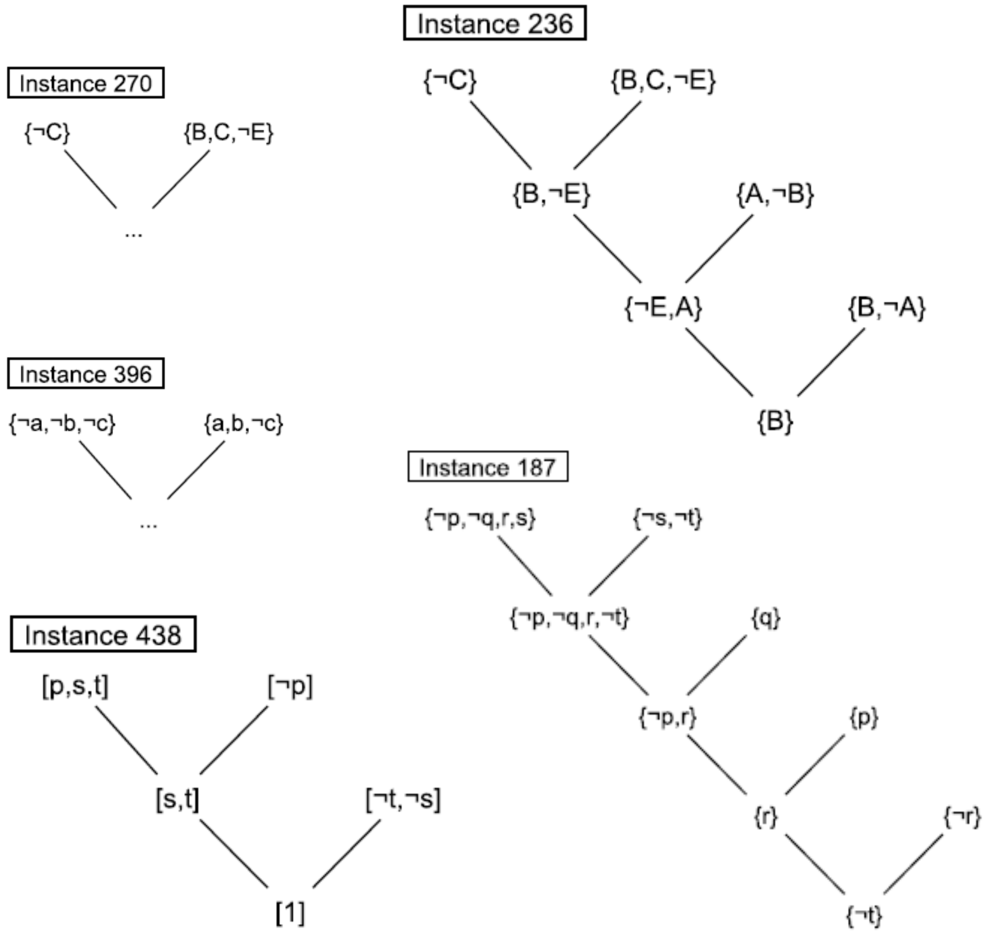
- 1/FO1($p$) … proximity of $p$ to elements from the same class
- FO2($p$) …… frequency of incorrect classification of $p$
- FO3($p$) ……. proximity $p$ to all elements, i.e. when ignoring the class attribute

# Outlier detection by weka-peka

| Instance | Error E3 | Outlier score |
|---|---|---|
| 270 | no | 131.96 |
| 396 | no | 131.96 |
| 236 | no | 73.17 |
| 187 | no | 61.03 |
| 438 | yes | 54.43 |
| 389 | yes | 52.50 |
| 74 | yes | 15.91 |
| 718 | yes | 15.91 |

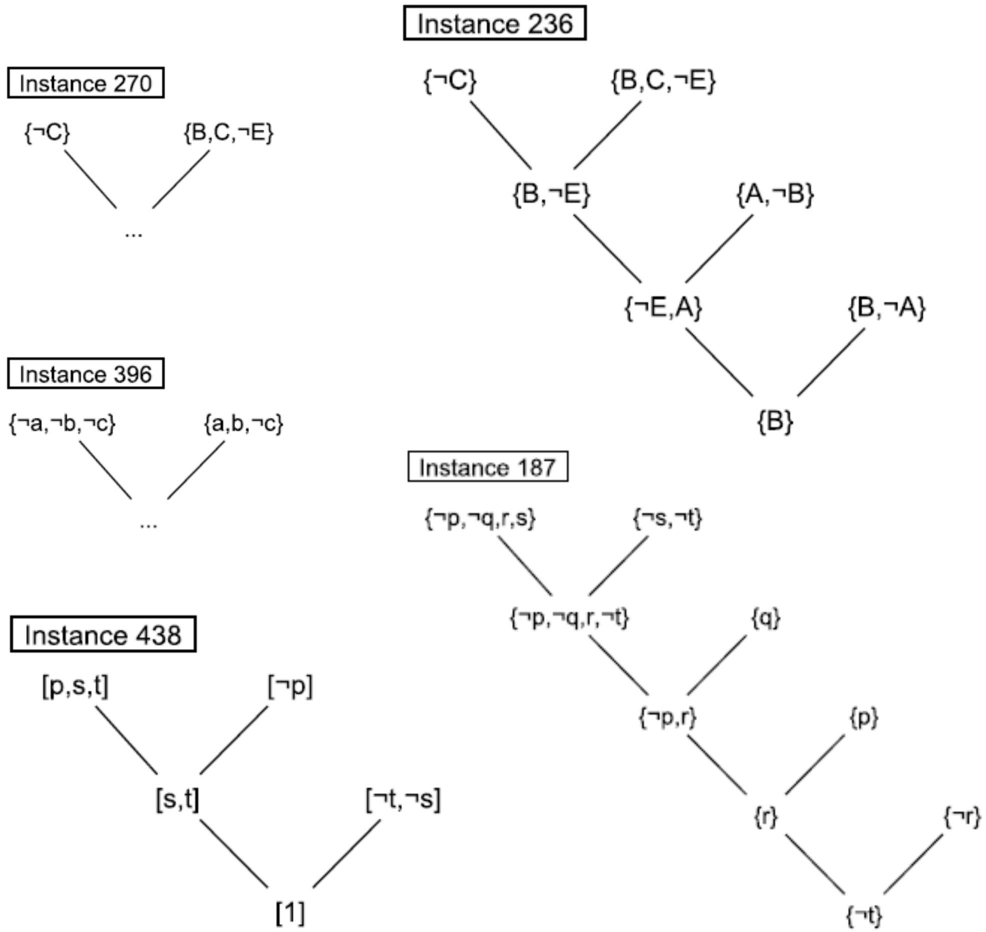
# Outlier Detection by weka-peka

| Instance | Error E3 | Outlier score |
|---|---|---|
| 270 | no | 131.96 |
| 396 | no | 131.96 |
| 236 | no | 73.17 |
| 187 | no | 61.03 |
| 438 | yes | 54.43 |
| 389 | yes | 52.50 |
| 74 | yes | 15.91 |
| 718 | yes | 15.91 |



# Outlier Detection by weka-peka

| Instance | Error E3 | Outlier score |
|---|---|---|
| 270 | no | 131.96 |
| 396 | no | 131.96 |
| 236 | no | 73.17 |
| 187 | no | 61.03 |
| 438 | yes | 54.43 |
| 389 | yes | 52.50 |
| 74 | yes | 15.91 |
| 718 | yes | 15.91 |




Goal now: to explain/interpret outliers

# References

VACULÍK, Karel, NEZVALOVÁ, Leona a Lubomír POPELÍNSKÝ. Graph Mining and Outlier Detection Meet Logic Proof Tutoring. Proc. of EDM 2014 Ws Graph-based Educational Data Mining

VACULÍK, Karel, Leona NEZVALOVÁ a Lubomír POPELÍNSKÝ. Educational data mining for analysis of students' solutions.16th International Conference, AIMSA 2014. London: Springer, 2014.

CODB:

Hewahi N.M. and Saad M.K. *Class Outliers Mining: Distance-Based Approach*. International Journal of Intelligent Systems and Technologies, Vol. 2, No. 1, pp 55-68, 2007.

pattern text

1 {} ; {neg Z,Z}

2 {} ; {}

3 {} ; {neg Z}

4 ...(cykli) ; ...(cykli)

5 {neg Z} ; {neg Y,Y}

6 {neg Y,Z} ; {neg X,X}

7 {} ; {Z}

8 {} ; {neg Y,neg Z,Y,Z}

9 {Y} ; {neg Y,neg Z,Z}

10 {} ; {Z,Z}

11 {neg Y} ; {neg Z,Y,Z}

12 {} ; {neg Z,neg Z}

13 {Z} ; {neg Y,Y}

14 {} ; {neg Y,neg Z,Y}

15 {} ; {neg Y,Y,Z,Z}

16 {Z} ; {neg X,neg Y,X,Y}

17 {neg Z} ; {neg X,neg Y,X,Y}

18 {} ; {neg X,neg Y,neg Z,X,Y,Z}

19 {Y} ; {neg X,X,Z}

20 {} ; {neg Y,Z}

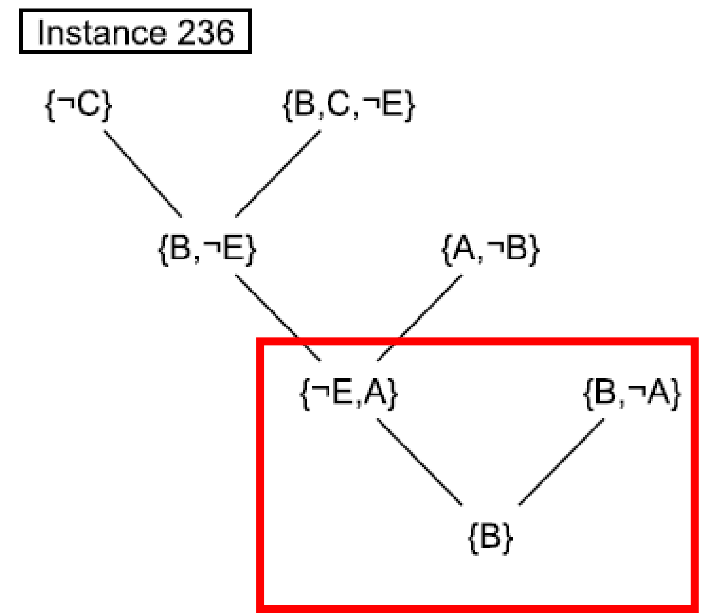
# AScore

- Let *P* be a pattern and suppose its value is *true* for the outlier
- Compute AScore of *P* as the fraction of remaining instances from the same class as the outlier whose value of *P* is equal to *false*
- Similarly for *P* equal to *false*, but AScore ranges from 0 to -1

# Outlier Explanation

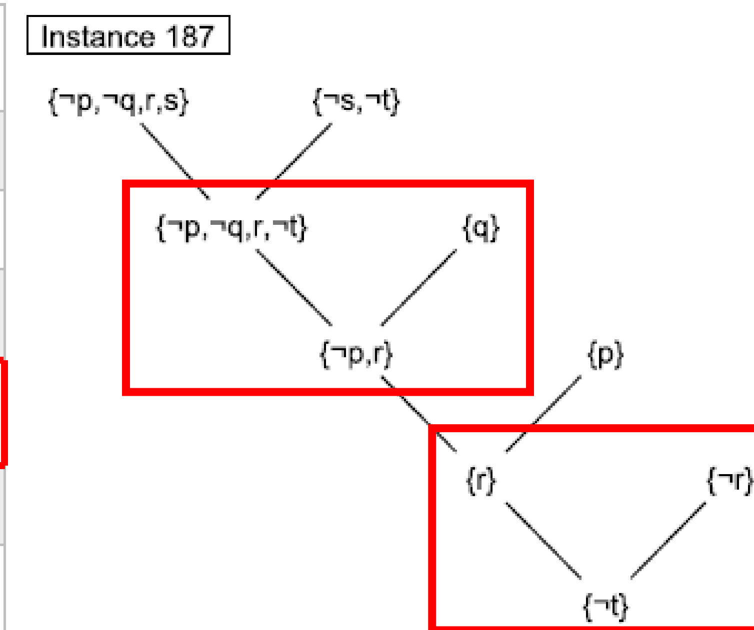
| Instance | Error E3 | Outlier score | Significant patterns* [(Ascore) *added; dropped*] | Significant missing patterns* [(Ascore) *added; dropped*] |
|---|---|---|---|---|
| 270 | no | 131.96 | (0.96) *looping* | (-0.99) {};{¬Z,Z} |
| 396 | no | 131.96 | (0.96) *looping* | (-0.99) {};{¬Z,Z} |
| 236 | no | 73.17 | (0.99) {};{¬Y, ¬Z,Y} | |
| 187 | no | 61.03 | (0.99) {¬Z};{¬Y,Y}<br>(0.99) {};{¬Y, ¬Z,Y} | |
| 438 | yes | 54.43 | (1.00) {Z};{¬X, ¬Y,X,Y} | (-0.94) {};{¬Y, ¬Z,Y,Z} |
| 389 | yes | 52.50 | (1.00) {};{¬Y, ¬Z,Y} | (-0.94) {};{¬Y, ¬Z,Y,Z}<br>(-0.81) {};{¬Z,Z} |
| 74 | yes | 15.91 | (0.98) {¬Z};{¬X, ¬Y,X,Y}<br>(0.98) {};{¬X, ¬Y, ¬Z,X,Y,Z} | (-0.94) {};{¬Y, ¬Z,Y,Z} |
| 718 | yes | 15.91 | (0.98) {¬Z};{¬X, ¬Y,X,Y}<br>(0.98) {};{¬X, ¬Y, ¬Z,X,Y,Z} | (-0.94) {};{¬Y, ¬Z,Y,Z} |

*Only patterns with |AScore| > 0.8 are displayed



# Outlier Explanation

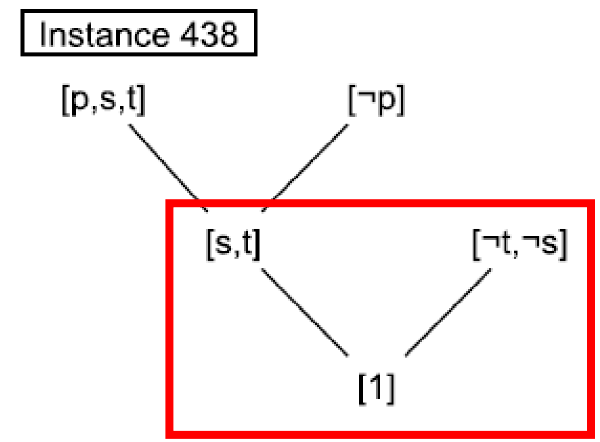
| Instance | Error E3 | Outlier score | Significant patterns* [(Ascore) *added; dropped*] | Significant missing patterns* [(Ascore) *added; dropped*] |
|---|---|---|---|---|
| 270 | no | 131.96 | (0.96) *looping* | (-0.99) {};{¬Z,Z} |
| 396 | no | 131.96 | (0.96) *looping* | (-0.99) {};{¬Z,Z} |
| 236 | no | 73.17 | (0.99) {};{¬Y, ¬Z,Y} | |
| 187 | no | 61.03 | (0.99) {¬Z};{¬Y,Y}<br>(0.99) {};{¬Y, ¬Z,Y} | |
| 438 | yes | 54.43 | (1.00) {Z};{¬X, ¬Y,X,Y} | (-0.94) {};{¬Y, ¬Z,Y,Z} |
| 389 | yes | 52.50 | (1.00) {};{¬Y, ¬Z,Y} | (-0.94) {};{¬Y, ¬Z,Y,Z}<br>(-0.81) {};{¬Z,Z} |
| 74 | yes | 15.91 | (0.98) {¬Z};{¬X, ¬Y,X,Y}<br>(0.98) {};{¬X, ¬Y, ¬Z,X,Y,Z} | (-0.94) {};{¬Y, ¬Z,Y,Z} |
| 718 | yes | 15.91 | (0.98) {¬Z};{¬X, ¬Y,X,Y}<br>(0.98) {};{¬X, ¬Y, ¬Z,X,Y,Z} | (-0.94) {};{¬Y, ¬Z,Y,Z} |




*Only patterns with |AScore| > 0.8 are displayed

# Outlier Explanation

| Instance | Error E3 | Outlier score | Significant patterns* [(Ascore) *added; dropped*] | Significant missing patterns* [(Ascore) *added; dropped*] |
|---|---|---|---|---|
| 270 | no | 131.96 | (0.96) *looping* | (-0.99) {};{¬Z,Z} |
| 396 | no | 131.96 | (0.96) *looping* | (-0.99) {};{¬Z,Z} |
| 236 | no | 73.17 | (0.99) {};{¬Y, ¬Z,Y} | |
| 187 | no | 61.03 | (0.99) {¬Z};{¬Y,Y}<br>(0.99) {};{¬Y, ¬Z,Y} | |
| 438 | yes | 54.43 | (1.00) {Z};{¬X, ¬Y,X,Y} | (-0.94) {};{¬Y, ¬Z,Y,Z} |
| 389 | yes | 52.50 | (1.00) {};{¬Y, ¬Z,Y} | (-0.94) {};{¬Y, ¬Z,Y,Z}<br>(-0.81) {};{¬Z,Z} |
| 74 | yes | 15.91 | (0.98) {¬Z};{¬X, ¬Y,X,Y}<br>(0.98) {};{¬X, ¬Y, ¬Z,X,Y,Z} | (-0.94) {};{¬Y, ¬Z,Y,Z} |
| 718 | yes | 15.91 | (0.98) {¬Z};{¬X, ¬Y,X,Y}<br>(0.98) {};{¬X, ¬Y, ¬Z,X,Y,Z} | (-0.94) {};{¬Y, ¬Z,Y,Z} |

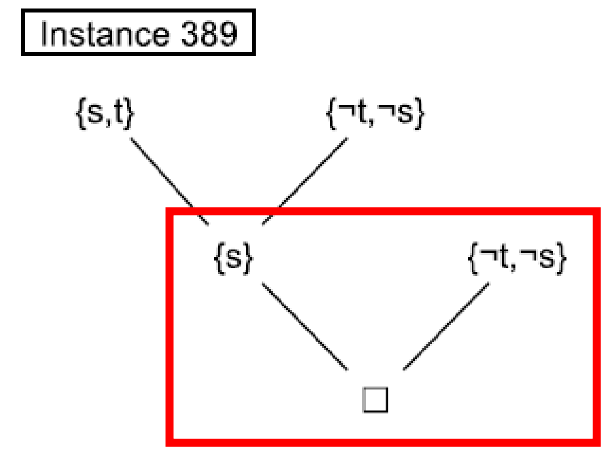



*Only patterns with |AScore| > 0.8 are displayed

# Outlier Explanation

| Instance | Error E3 | Outlier score | Significant patterns* [(Ascore) *added; dropped*] | Significant missing patterns* [(Ascore) *added; dropped*] |
|---|---|---|---|---|
| 270 | no | 131.96 | (0.96) *looping* | (-0.99) {};{¬Z,Z} |
| 396 | no | 131.96 | (0.96) *looping* | (-0.99) {};{¬Z,Z} |
| 236 | no | 73.17 | (0.99) {};{¬Y, ¬Z,Y} | |
| 187 | no | 61.03 | (0.99) {¬Z};{¬Y,Y}<br>(0.99) {};{¬Y, ¬Z,Y} | |
| 438 | yes | 54.43 | (1.00) {Z};{¬X, ¬Y,X,Y} | (-0.94) {};{¬Y, ¬Z,Y,Z} |
| 389 | yes | 52.50 | (1.00) {};{¬Y, ¬Z,Y} | (-0.94) {};{¬Y, ¬Z,Y,Z}<br>(-0.81) {};{¬Z,Z} |
| 74 | yes | 15.91 | (0.98) {¬Z};{¬X, ¬Y,X,Y}<br>(0.98) {};{¬X, ¬Y, ¬Z,X,Y,Z} | (-0.94) {};{¬Y, ¬Z,Y,Z} |
| 718 | yes | 15.91 | (0.98) {¬Z};{¬X, ¬Y,X,Y}<br>(0.98) {};{¬X, ¬Y, ¬Z,X,Y,Z} | (-0.94) {};{¬Y, ¬Z,Y,Z} |

*Only patterns with |AScore| > 0.8 are displayed



# Sapling Random Forest (LOF)

- Train trees (sapling) to distinguish normal instances from outliers
- Only subset of normal instances are used (k-nereast neighbours vs. random)
- Find explanation in resulting paths
- Knowledge about classes not taken into account

# Tree Reduction

- Original classes replaced with two new classes
  - Normal class – includes labeled class of given instance
  - Outlier class – includes others original classes
- Analysis of the trees that classify instance to Outlier class
- Change attribut values on the path in given tree for given instance and check if resulting class change → attribut is part of explanation
- Find explanation in resulting paths and calculate support

# Results

| Instance | Error E3 | Outlier score | AScore | Sapling Random Forest (LOF) | Tree Reduction |
|---|---|---|---|---|---|
| 270 | no | 131.96 | pattern_1 (-0.99)<br>pattern_4 (0.96) | pattern_1=0 (1) | pattern_1=0 (0.81) |
| 396 | no | 131.96 | pattern_1 (-0.99)<br>pattern_4 (0.96) | pattern_1=0 (1) | pattern_1=0 (0.81) |
| 236 | no | 73.17 | pattern_14 (0.99) | pattern_2=0 (0.8) | pattern_14=1 (0.46)<br>pattern_14=1 && pattern_5=0 (0.22) |
| 187 | no | 61.03 | pattern_14 (0.99)<br>pattern_5 (0.99) | pattern_2=0 (0.95) | pattern_14=1 (0.76) |
| 438 | yes | 54.43 | pattern_16 (1.00)<br>pattern_8 (-0.94) | pattern_2=0 (0.75) | pattern_1=1 && pattern_8=0 && pattern_17=0 (0.23)<br>pattern_8=0 && pattern_17=0 (0.20) |
| 389 | yes | 52.50 | pattern_14 (1.00)<br>pattern_8 (-0.94)<br>pattern_1 (-0.81) | pattern_1=0 (1) | pattern_8=0 (0.26)<br>pattern_8=0 && pattern_16=0 && pattern_17=0 (0.21)<br>pattern_8=0 && pattern_16=0 && pattern_18=0 (0.21) |
| 74 | yes | 15.91 | pattern_17 (0.98)<br>pattern_18 (0.98)<br>pattern_8 (-0.94) | pattern_2=0 (0.7) | pattern_8=0 && pattern_16=0 (0.29)<br>pattern_1=1 && pattern_8=0 && pattern_16=0 (0.26) |