Conformal Association Rule Mining (CARM)

Ilia Nouretdinov, James Gammerman,
Centre for Reliable Machine Learning,
Royal Holloway, University of London, Egham, Surrey, UK

Royal Holloway, University of London

i.r.nouretdinov@rhul.ac.uk

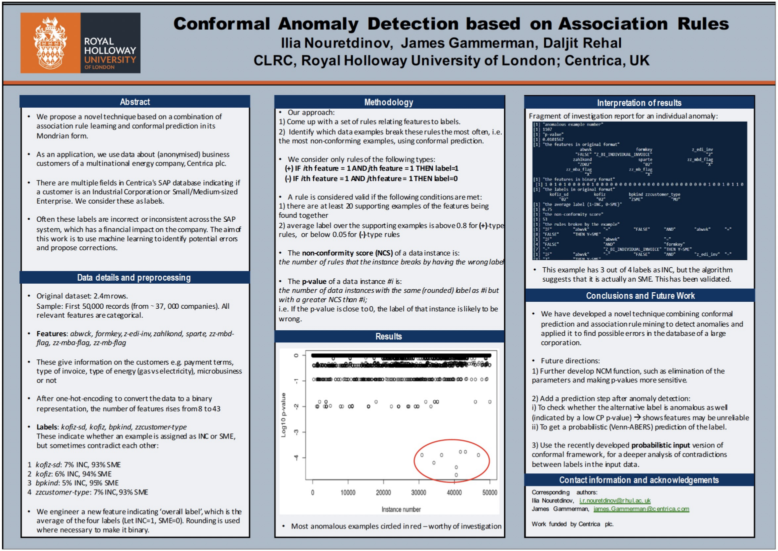
jgammerman@gmail.com

September 14, 2023

Introduction

- CARM is an integration of two frameworks: *conformal prediction* and *association rule mining*.
- It enables detection of errors within a set of binary labels, with the usual CP guarantees on validity.
- XAI angle
- As an extension, we analyse the errors using *probabilistic prediction* to suggest corrections.

Origin story: COPA 2019 poster



Conformal Anomaly Detection based on Association Rules

Ilia Nouretdinov, James Gammerman, Daljit Rehal
CLRC, Royal Holloway University of London; Centrica, UK

Abstract

- We propose a novel technique based on a combination of association rule learning and conformal prediction in its Mondrian form.
- As an application, we use data about (anonymised) business customers of a multinational energy company, Centrica plc.
- There are multiple fields in Centrica's SAP database indicating if a customer is an Industrial Corporation or Small/Medium-sized Enterprise. We consider these as labels.
- Often these labels are incorrect or inconsistent across the SAP system, which has a financial impact on the company. The aim of this work is to use machine learning to identify potential errors and propose corrections.

Data details and preprocessing

- Original dataset: 2.4m rows.
  Sample: First 50,000 records (from ~ 37, 000 companies). All relevant features are categorical.
- **Features**: *abwck, formkey, z-edi-inv, zahlkond, sparte, zz-mbd-flag, zz-mba-flag, zz-mb-flag*
- These give information on the customers e.g. payment terms, type of invoice, type of energy (gas vs electricity), microbusiness or not
- After one-hot-encoding to convert the data to a binary representation, the number of features rises from 8 to 43
- **Labels**: *kofiz-sd, kofiz, bpkind, zzcustomer-type*
  These indicate whether an example is assigned as INC or SME, but sometimes contradict each other:

1 *kofiz-sd*: 7% INC, 93% SME
2 *kofiz*: 6% INC, 94% SME
3 *bpkind*: 5% INC, 95% SME
4 *zzcustomer-type*: 7% INC, 93% SME

- We engineer a new feature indicating 'overall label', which is the average of the four labels (Let INC=1, SME=0). Rounding is used where necessary to make it binary.

Methodology

- Our approach:

1) Come up with a set of rules relating features to labels.
2) Identify which data examples break these rules the most often, i.e. the most non-conforming examples, using conformal prediction.

- We consider only rules of the following types:

(+) IF *i*th feature = 1 AND *j*th feature = 1 THEN label=1
(-) IF *i*th feature = 1 AND *j*th feature = 1 THEN label=0

- A rule is considered valid if the following conditions are met:

1) there are at least 20 supporting examples of the features being found together
2) average label over the supporting examples is above 0.8 for **(+)**-type rules, or below 0.05 for **(-)**-type rules

- The **non-conformity score (NCS)** of a data instance is:

the number of rules that the instance breaks by having the wrong label

- The **p-value** of a data instance *#i* is:

the number of data instances with the same (rounded) label as #i but with a greater NCS than #i;
i.e. If the p-value is close to 0, the label of that instance is likely to be wrong.

Results

- Most anomalous examples circled in red – worthy of investigation

Interpretation of results

Fragment of investigation report for an individual anomaly:

```
[1] "anomalous example number"
[1] 1107
[1] "p-value"
[1] 0.0101567
[1] "the features in original format"
                  abwvk                formkey              z_edi_inv
                "FALSE" "Z_BI_INDIVIDUAL_INVOICE"                  "2"
               zahlkond                 sparte             zz_mbd_flag
                 "ZD02"                   "02"                   "X"
            zz_mba_flag              zz_mb_flag
                    "X"                    "X"
[1] "the features in binary format"
 [1] 1 0 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 1 1 0
[1] "the labels in original format"
       kofiz_sd          kofiz        bpkind zzcustomer_type
           "02"           "02"        "ZSME"            "MU"
[1] "the average label (1=INC, 0=SME)"
[1] 0.75
[1] "the non-conformity score"
[1] 53
[1] "the rules broken by the example"
[1] "IF"         "abwvk"      "="          "FALSE"      "AND"        "abwvk"      "="
[8] "FALSE"      "THEN Y=SME"
[1] "IF"                      "abwvk"                   "="
[4] "FALSE"                   "AND"                     "formkey"
[7] "="                       "Z_BI_INDIVIDUAL_INVOICE" "THEN Y=SME"
[1] "IF"         "abwvk"      "="          "FALSE"      "AND"        "z_edi_inv"  "="
[8] "2"          "THEN Y=SME"
```

- This example has 3 out of 4 labels as INC, but the algorithm suggests that it is actually an SME. This has been validated.

Conclusions and Future Work

- We have developed a novel technique combining conformal prediction and association rule mining to detect anomalies and applied it to find possible errors in the database of a large corporation.
- Future directions:

1) Further develop NCM function, such as elimination of the parameters and making p-values more sensitive.

2) Add a prediction step after anomaly detection:
i) To check whether the alternative label is anomalous as well (indicated by a low CP p-value) → shows features may be unreliable
ii) To get a probabilistic (Venn-ABERS) prediction of the label.

3) Use the recently developed **probabilistic input** version of conformal framework, for a deeper analysis of contradictions between labels in the input data.

Contact information and acknowledgements

Corresponding authors:
Ilia Nouretdinov, i.r.nouretdinov@rhul.ac.uk
James Gammerman, james.Gammerman@centrica.com

Work funded by Centrica plc.

Association Rule Mining

- A rule-based ML method for discovering relationships between items in a dataset.
- E.g for supermarkets: {bread} → {butter}
- In general terms: *{antecedent}* → *{consequent}*
- In a ML setting, an example rule might be:

  $$\text{IF feature } F = a \text{ THEN feature } G = b$$

  G could alternatively refer to a label.
- **Implication**: Given a set of rules, errors can be identified as deviations from rules that hold true for most of the data

ARM concepts

- **Support**: the proportion of examples where the rule holds.
- **Confidence**: the conditional probability of the rule's consequent given its antecedent.
- Most common way to generate rules: *apriori algorithm*
  - Requires user-defined thresholds for minimum support and confidence

Apriori algorithm: lack of statistical rigor

- No notion of statistical significance

- Arbitrary choice of thresholds can lead to spurious and missed rules

- Solution: use a different approach to ARM...

Our approach to ARM

- As per Hamalainen et al. (2009)
- Idea: subject any possible rule to a binomial test
- Each example is treated as an independent Bernoulli trial, whose outcome is either 1 or 0
  - 1 means the rule $(F = a) \to (G = b)$ occurs
  - 0 means the rule doesn't occur
- H_0: no association between antecedent and consequent
- Interpretation of p-values: the ‘weight’ of a given rule
  - a small value provides evidence against H_0

Non-conformity measure

- So we have...
  - A complete pool of rules with associated weights p_r
  - A set of examples Z which either comply with those rules or not
- So for a given example $z \in Z$, we have an expectation for its label based on each rule r (assuming the antecedent of r is true for z)
- If the expected label doesn't match the observed label, then z is an exception to that rule, i.e. non-conforming.
- Non-conformity score α for z:
  - Initialise α at 0, then add $-\log p_r$ for every broken rule
  - accumulates degree of non-conformity

Converting NCM to conformal p-values

- NB. Different kind of p-values!
- **Input**: The data sequence $Z' = ((x_1, y_1), \ldots, (x_N, y_N))$, a non-conformity measure $\mathcal{A}$, and a threshold ε.
- **Non-conformity scores**: For each data point $i = 1, \ldots, N$, compute:

$$\alpha_i = \mathcal{A}\left((x_i, y_i), Z' \setminus \{(x_i, y_i)\}\right)$$

- ***p*-value Calculation**: Determine p_N using:

$$p_N = \frac{\left|\left\{j : y_j = y_N, \alpha_j \geq \alpha_N\right\}\right|}{\left|\left\{j : y_j = y_N\right\}\right|}$$

- **Error Detection**: If $p_N < \varepsilon$, flag the example z_N as an error.

Validity

- All conformal predictors automatically come with the property of *validity*, i.e. the probability of incorrectly rejecting H_0 is at most ε.

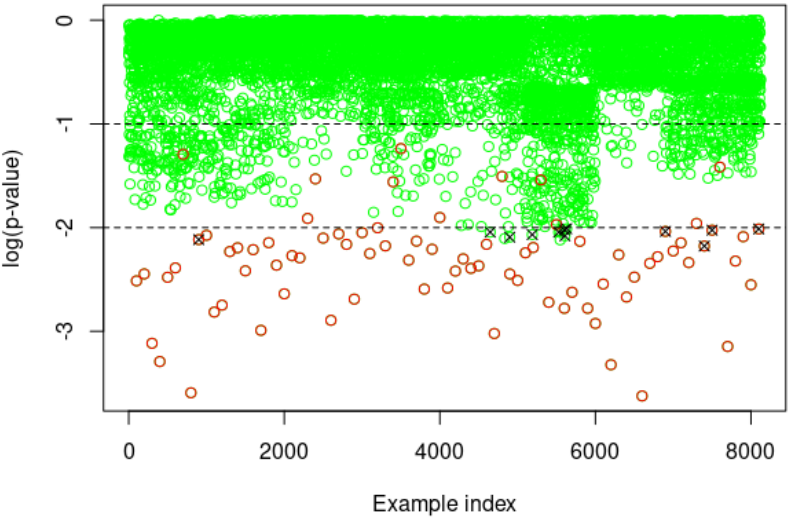
- In our setting of error detection, ε becomes an upper bound on the probability of a false positive (false alarm).

Datasets & pre-processing

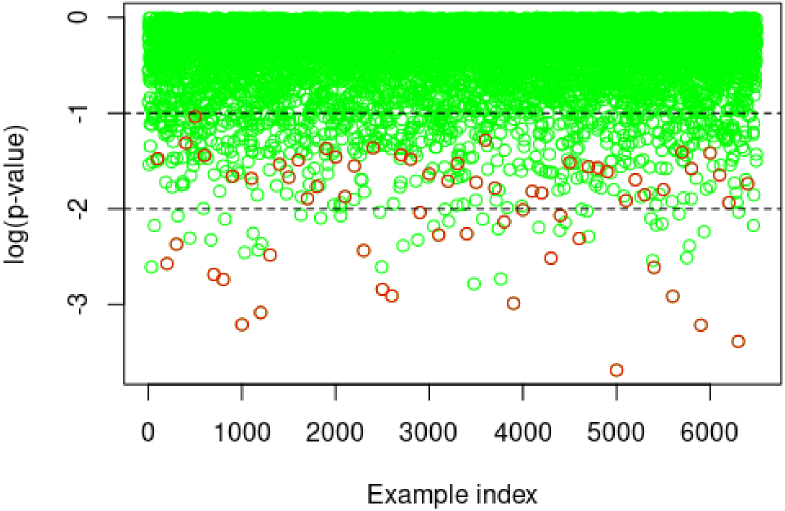
1. **Mushrooms.** Label: edible or posionous
2. **Wine Quality.** Label: red or white
3. **Adult Income.** Label: salary greater/less than $50K

- Introduced known errors into the labels for 1% of examples
- Remaining features converted to binary form by one-hot encoding

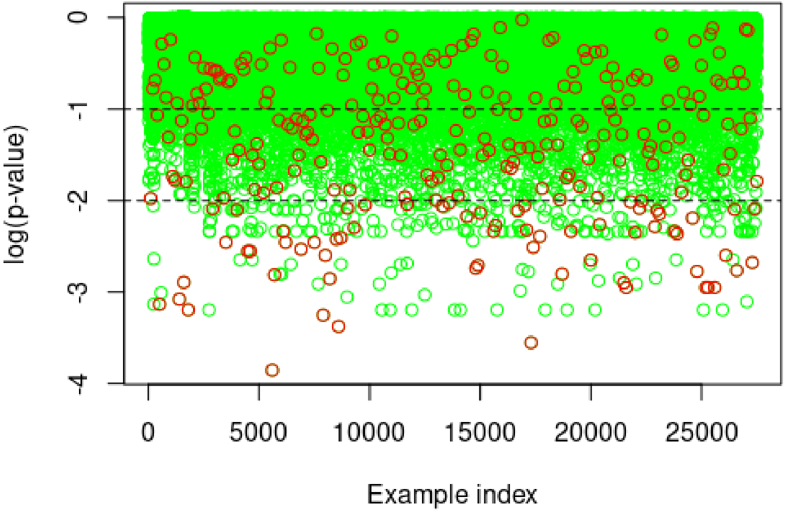
CARM *p*-values (log scale) - Mushrooms



CARM *p*-values (log scale) - Wine



CARM *p*-values (log scale) - Adult



Evaluation of CARM

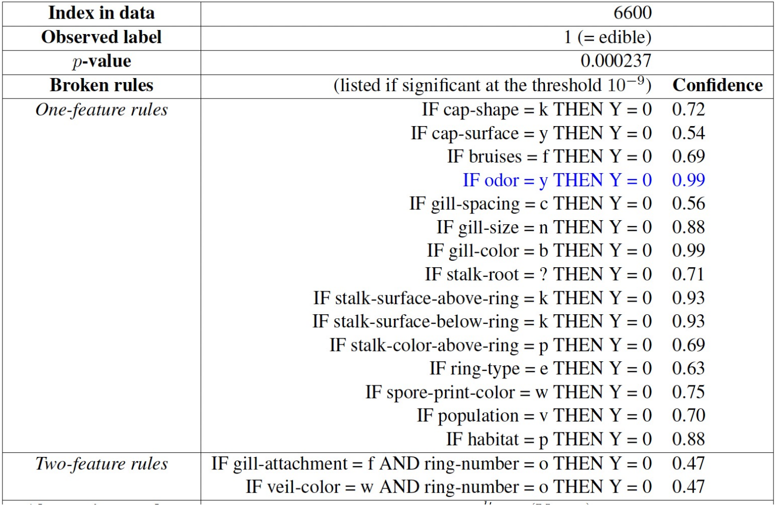
Precision at three *p*-value thresholds

Dataset	$p < 0.1\%$	$p < 1\%$	$p < 10\%$
Mushrooms	100%	87%	10%
Wine	100%	37%	10%
Adult Income	29%	23%	6%

Recall at three *p*-value thresholds

Dataset	$p < 0.1\%$	$p < 1\%$	$p < 10\%$
Mushrooms	9%	87%	100%
Wine	8%	37%	100%
Adult Income	2%	23%	62%

Example report



Index in data	6600	
Observed label	1 (= edible)	
p**-value**	0.000237	
Broken rules	(listed if significant at the threshold 10^{-9})	**Confidence**
One-feature rules	IF cap-shape = k THEN Y = 0	0.72
	IF cap-surface = y THEN Y = 0	0.54
	IF bruises = f THEN Y = 0	0.69
	IF odor = y THEN Y = 0	0.99
	IF gill-spacing = c THEN Y = 0	0.56
	IF gill-size = n THEN Y = 0	0.88
	IF gill-color = b THEN Y = 0	0.99
	IF stalk-root = ? THEN Y = 0	0.71
	IF stalk-surface-above-ring = k THEN Y = 0	0.93
	IF stalk-surface-below-ring = k THEN Y = 0	0.93
	IF stalk-color-above-ring = p THEN Y = 0	0.69
	IF ring-type = e THEN Y = 0	0.63
	IF spore-print-color = w THEN Y = 0	0.75
	IF population = v THEN Y = 0	0.70
	IF habitat = p THEN Y = 0	0.88
Two-feature rules	IF gill-attachment = f AND ring-number = o THEN Y = 0	0.47
	IF veil-color = w AND ring-number = o THEN Y = 0	0.47

Probabilistic Prediction

- When a data example is identified as an error, it is important to conduct a thorough investigation that includes also a suggested correction for it.
- For this purpose, we can employ the *Venn-ABERS (VA) framework*.

- VA works in same assumptions as CP.
- Like CP, VA framework can be linked to an underlying method as well.
- The difference is in output: where CP produces *p-values*, VA outputs *lower* and *upper probabilities*.

CARM & VA results for 10 most suspicious examples

Index	Label.	*p*-value	VA probs	VA prediction	True label	Comment
6600	1	0.00024	0.0074–0.0078	0	0	1 corrected to 0
800	0	0.00026	0.990–0.991	1	1	0 corrected to 1
6200	1	0.00048	0.0074 – 0.0078	0	0	1 corrected to 0
400	0	0.00051	0.990 – 0.991	1	1	0 corrected to 1
7700	1	0.00071	0.0074 – 0.0078	0	0	1 corrected to 0
300	0	0.00077	0.990 – 0.991	1	1	0 corrected to 1
4700	1	0.00095	0.00780 – 0.98	0	0	1 corrected to 0
1700	0	0.001	0.990 – 0.991	1	1	0 corrected to 1
6000	1	0.0012	0.00737 – 0.00780	0	0	1 corrected to 0
2600	0	0.0013	0.990 – 0.991	1	1	0 corrected to 1

- *p*-values for alternative labels are all close to 1, i.e. no longer suspicious.
- VA predictions indicate high confidence in alternative labels.

Conclusions and further work

- Demonstrated integration of modified ARM with the CP framework for *explainable* error detection in data labels.
- Validity property limits false alarms during error detection.
- Association rules enhances interpretability and serve as basis for probabilistic analysis using Venn-ABERS

Further work:

- Exploration of more complex rules.
- Extension of the methodology for multi-class labels
- Extension of methodology to correcting features, not just labels