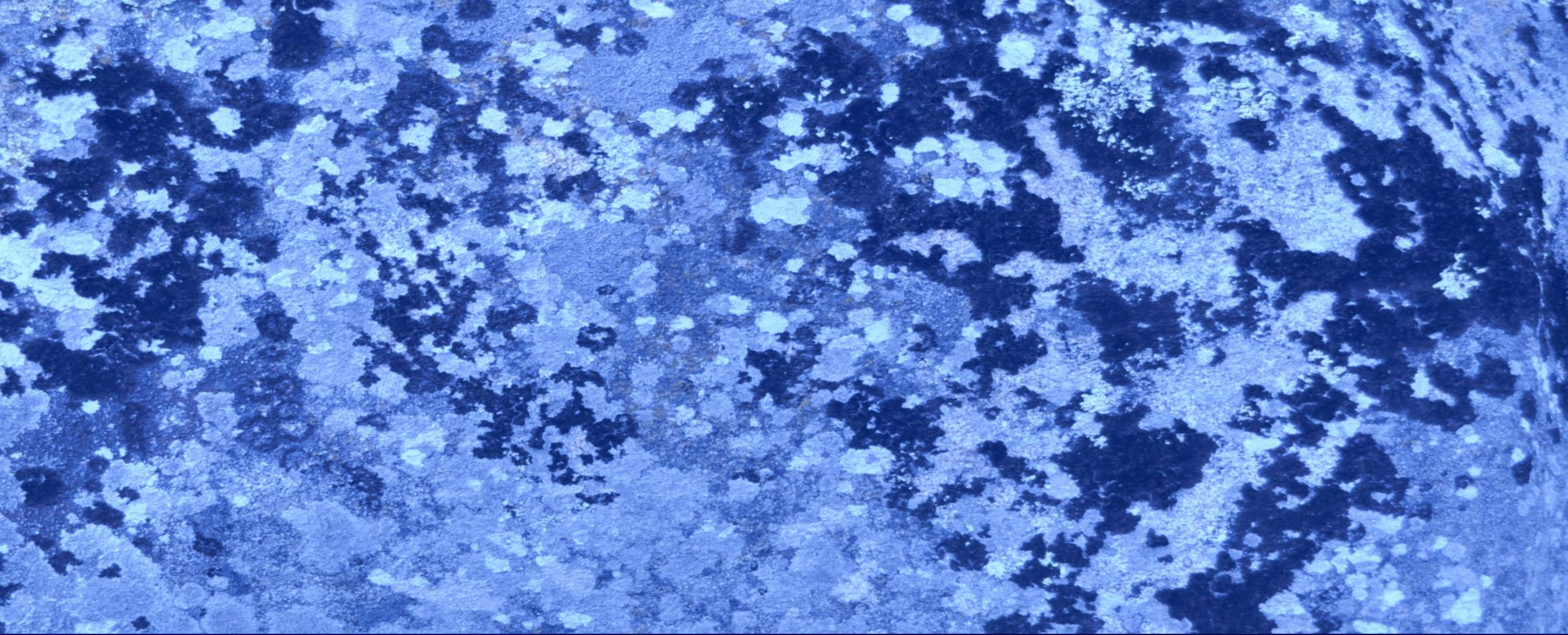


dr. Ralph Foorthuis

The SECODA Algorithm for the Detection of Anomalies in Sets with Mixed Data

Introduction

- Ralph Foorthuis

  Lead architect data domain at UWV

- This presentation gives an overview of several publications:
  - *A Typology of Data Anomalies*, Proceedings of IPMU 2018
  - *SECODA: Segmentation- and Combination-Based Detection of Anomalies*, Proceedings of IEEE DSAA 2017
  - *Anomaly Detection with SECODA*, Poster Presentation at IEEE DSAA 2017
  - Data examples and resources for R at www.foorthuis.nl

SECODA

- A novel general-purpose anomaly detection algorithm
  - Unsupervised
  - Non-parametric
  - Allows analysis of mixed data (numerical & categorical)
- The method is guaranteed to identify cases with unique or sparse combinations of attribute values.

Anomaly detection

Anomaly detection (AD) aims at identifying data cases that are in some way awkward and do not appear to fit the general pattern(s) present in the dataset.

AD is useful for fraud detection, data quality analysis, security scanning, data cleansing/modelling, system monitoring, etc.

Several types of anomalies can be acknowledged.

Related research

- A substantive body of research on AD is available [1, 7, 14, 23].
- AD originally involved *statistical parametric methods* that focus on univariate outliers [3, 16].
- *Non-parametric multidimensional distance-based methods* [8, 24] focus on the distance between individual data points and their nearest neighbors. This has been advanced throughout the years in order to also take into account larger datasets as well as categorical attributes [18, 19, 20, 21, 25].
- *Density-based approaches* focus on the amount of data points in each point's neighborhood [4, 5, 26]. Anomalies are found in low-density areas. The *histogram-based technique* is one of the traditional methods.
- Examples of *complex non-parametric statistical models* for AD are One-Class Support Vector Machines [27], ensembles [28, 29] and various subspace methods [1, 30, 49].

Typology of Anomalies

- Gives an overview of the types of anomalies.
- Provides a theoretical and tangible *understanding* of the types of anomalies.
- Aids in *evaluating* which types of anomalies can be detected by a given AD algorithm.
- Assists in *analyzing* conceptual levels of patterns and anomalies, and comparing between typologies.

Dimensions of the typology

Differentiates between the ‘awkward cases’ according to two fundamental dimensions regarding data:

- The data types that describe the behavior of the cases: numeric, categorical or both.
- The cardinality of relationship: whether anomalous behavior should be attributed to individual and independent variables (univariate) or to the relationship between variables (multivariate).

These dimensions naturally and objectively yield six basic types of anomalies.

Typology of Anomalies

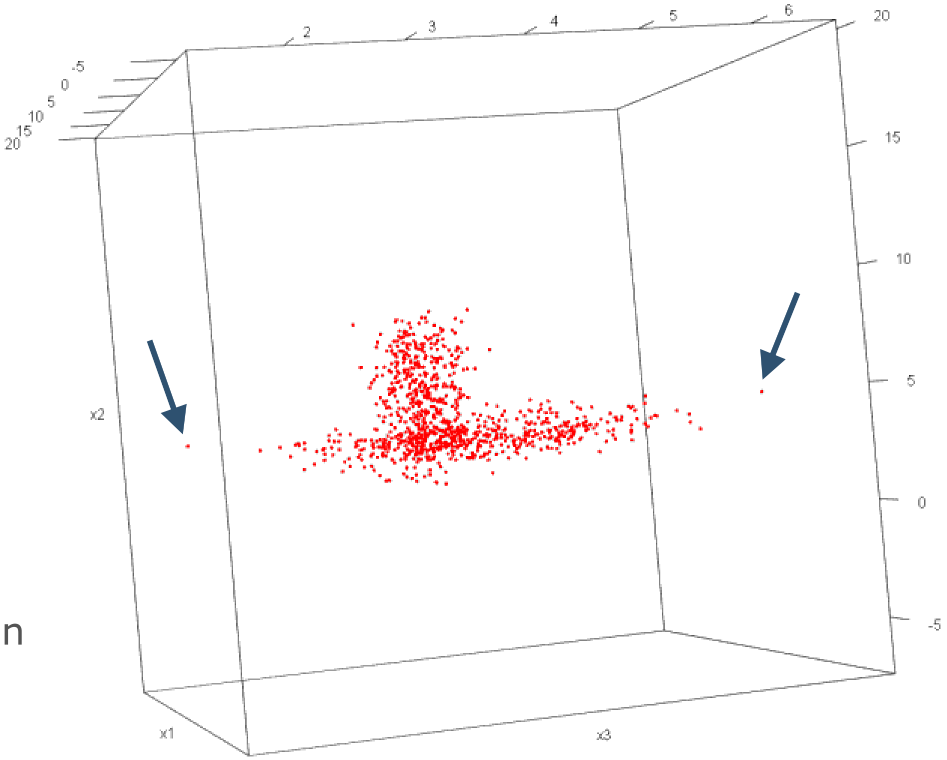
		Types of data		
		Continuous attributes	**Categorical attributes**	**Mixed attributes**
Cardinality of Relationship	**Univariate** Described by individual attributes (independence)	Type I Extreme value anomaly	Type II Rare class anomaly	Type III Simple mixed data anomaly
	Multivariate Described by multidimensionality (dependence)	Type IV Multidimensional numerical anomaly	Type V Multidimensional rare class anomaly	Type VI Multidimensional mixed data anomaly

From: Foorthuis (2018), ‘A Typology of Data Anomalies’, IPMU 2018.

Type I - Extreme value anomaly

- A case with an extremely high, low or otherwise rare value for one or multiple individual numerical attributes [cf. 1, 3, 56, 57].
- Such a case has one or more numerical values that can be considered extreme when the entire dataset is taken into account.
- Univariate: Can be identified by focusing on individual attributes. There is no need to analyze attributes jointly. However, a case is more anomalous if it has multiple attributes with extreme values.
- Traditional univariate statistics typically considers this type of outlier, e.g. by using a measure of central tendency plus or minus 3 times the standard deviation.

Type I - Extreme value anomaly

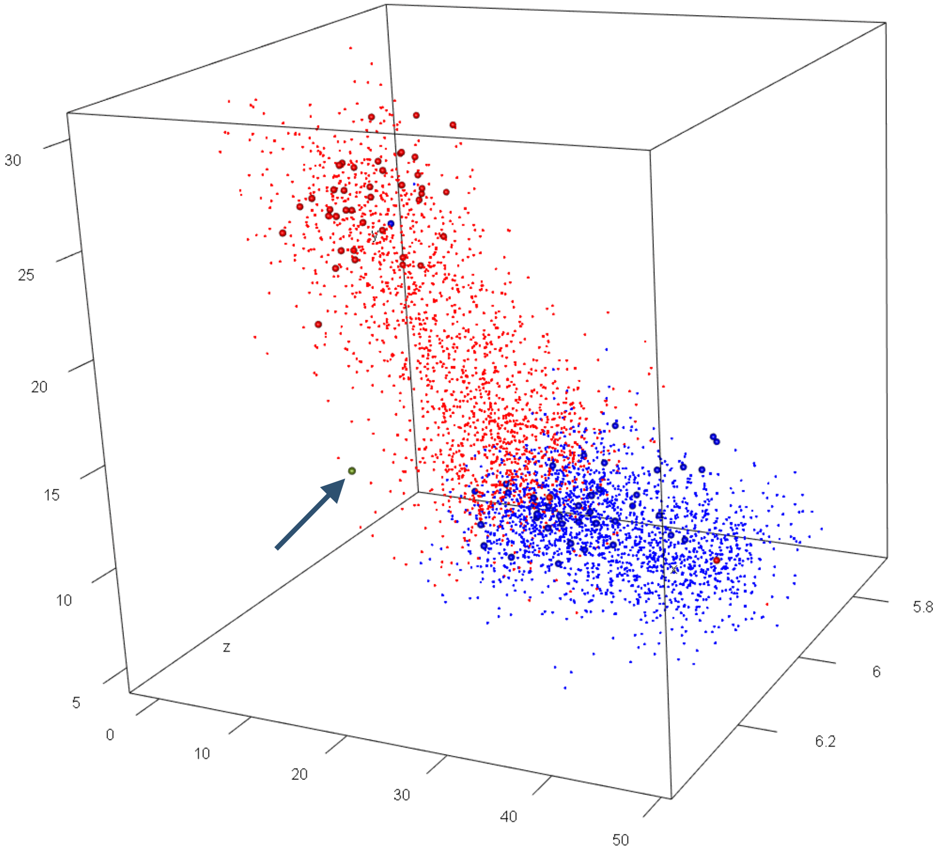


Extreme value on
the *x3* variable

Type II - Sparse class anomaly

- A case with a rare class value on one or multiple categorical attributes [cf. 17, 56, 57].
- Such a case has one or more class values that can be considered rare when the entire dataset is taken into account.
- Univariate: Can be identified by focusing on individual attributes. There is no need to analyze attributes jointly. However, a case is more anomalous if it has multiple attributes with sparse classes.

Type II - Sparse class anomaly

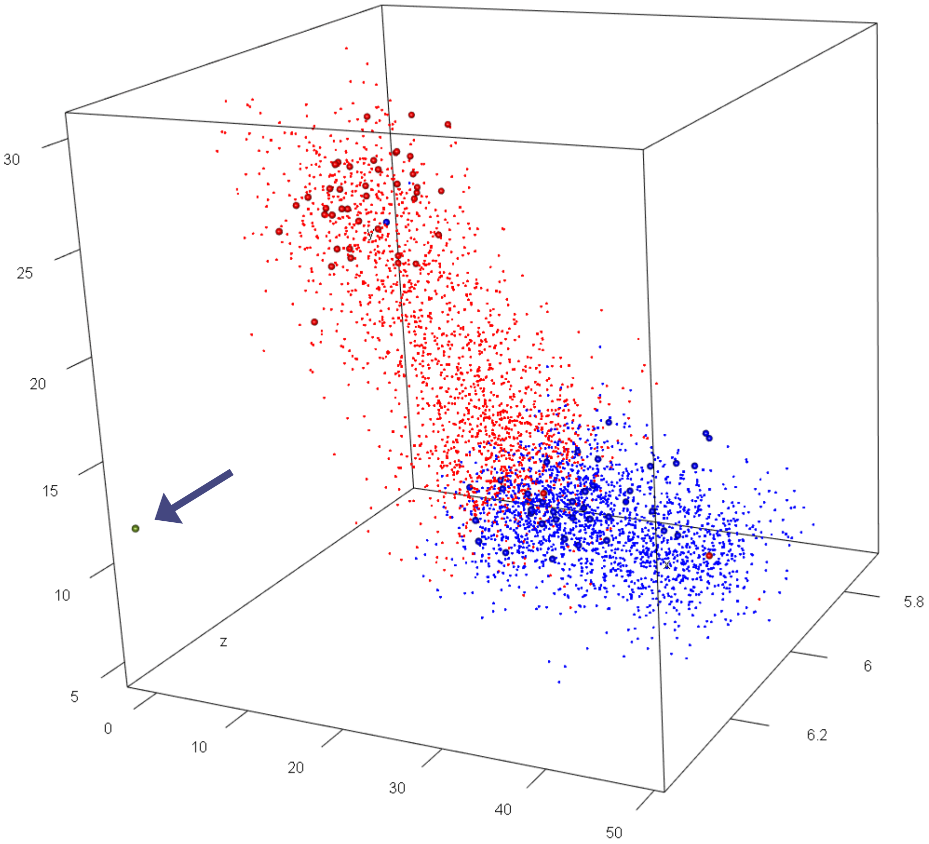


The only data point
of class “green”

Type III – Simple mixed data anomaly

- A case that is both a Type I and Type II anomaly, i.e. with at least one extreme value and one rare class.
- This anomaly type deviates with regard to multiple data types.
- Requires deviant values for at least two attributes, each anomalous in its own right.
- Univariate: Can be identified by focusing on the individual attributes.

Type III – Simple mixed data anomaly



The only data point of class “green”

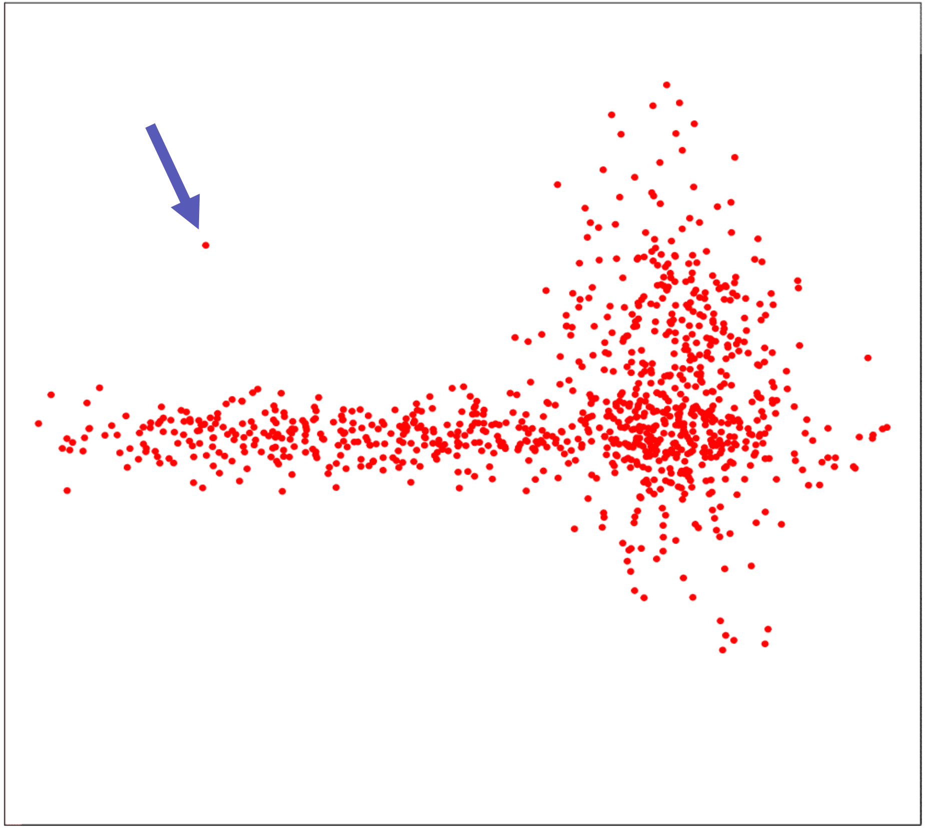
and

At the extreme left

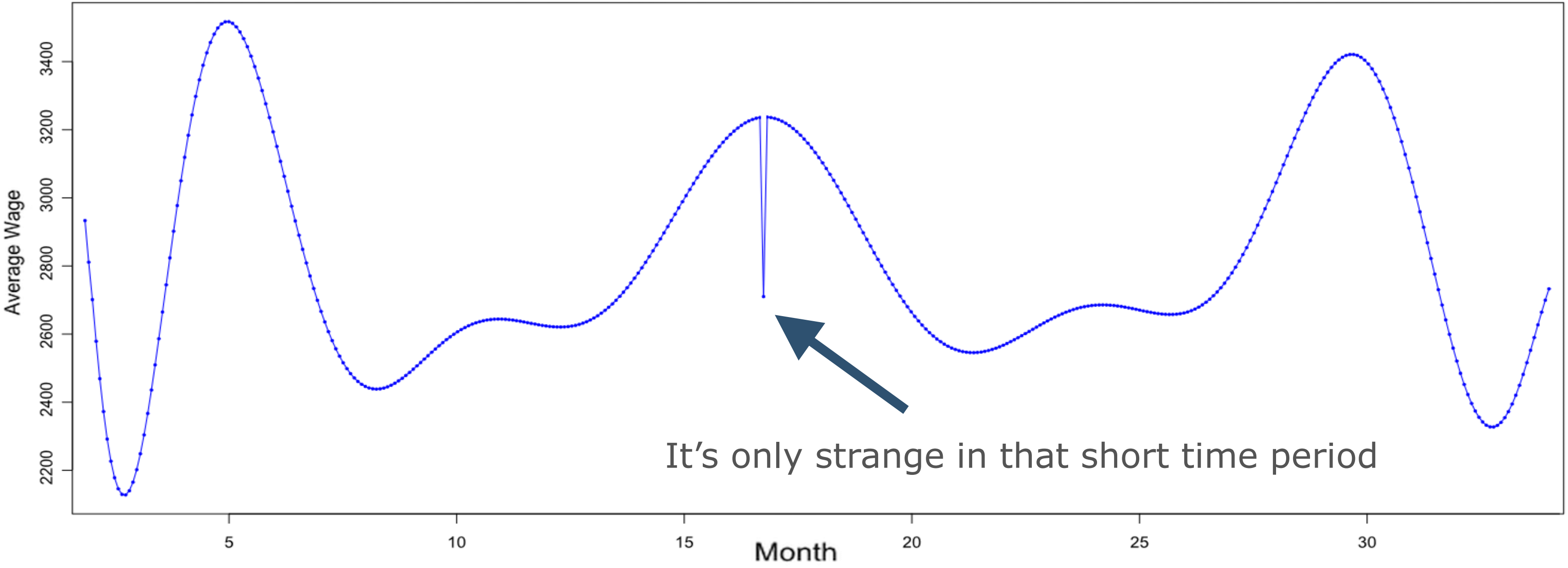
Type IV - Multidimensional numerical anomaly

- A case that does not conform to the general pattern when multiple numerical attributes are taken into account, but does not feature extreme values for any of its individual numerical attributes [cf. 14, 15, 56, 57].
- Multivariate: Focuses on multiple attributes. Such cases hide in multidimensionality [cf. 2], so several attributes have to be analyzed jointly to detect that they are located in an isolated area.

Type IV - Multidimensional numerical anomaly



Type IV - Multidimensional numerical anomaly

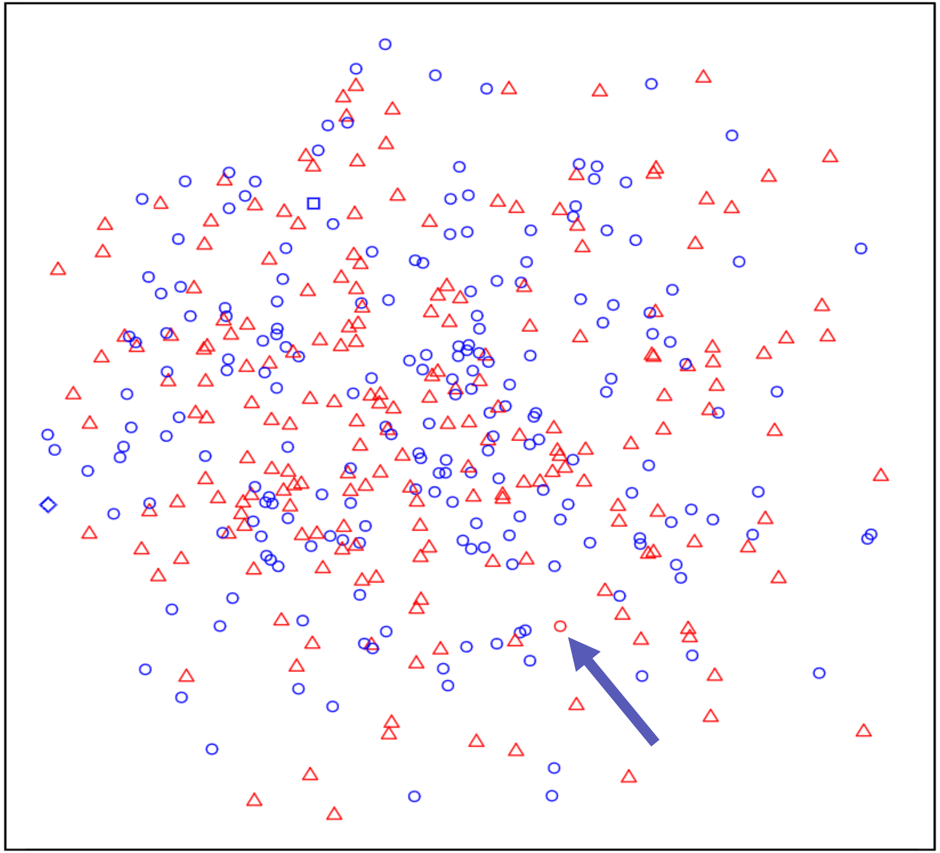


It’s only strange in that short time period

Type V - Multidimensional rare class anomaly

- A case with a rare combination of class values.
- A minimum of two substantive categorical attributes needs to be analyzed jointly to discover a multidimensional rare class anomaly.
- An example is this curious combination of values from three attributes used to describe dogs: ‘MALE’, ‘PUPPY’ and ‘PREGNANT’.
- In datasets with dependent data points the class values can be from one substantive attribute.

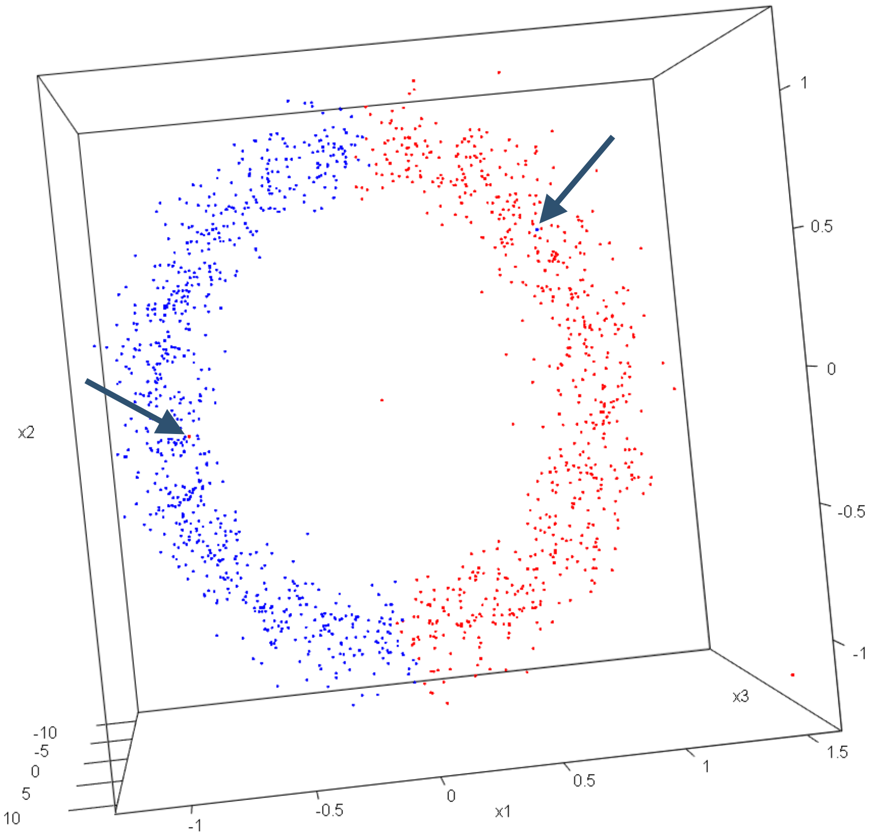
Type V - Multidimensional rare class anomaly



Type VI - Multidimensional mixed data anomaly

- A case with a categorical value or a combination of categorical values that in itself is not rare in the dataset as a whole, but is only rare in its neighborhood (numerical area) or local pattern [56, 57].
- Multivariate: Focuses on multiple attributes. Such cases hide in multidimensionality and multiple attributes need thus to be taken into account jointly to identify them.
- In fact, multiple datatypes need to be used, as a type IV anomaly per definition contains both numerical and categorical data.
- Cases can also take the form of second- or higher-order anomalies, with categorical values that are not rare (not even in their neighborhood), but prove to be rare in their combination in that specific area. See the DSAA paper for an example.

Type VI - Multidimensional mixed data anomaly



Some points seem to be misplaced, having the wrong color in their neighborhood

The SECODA algorithm

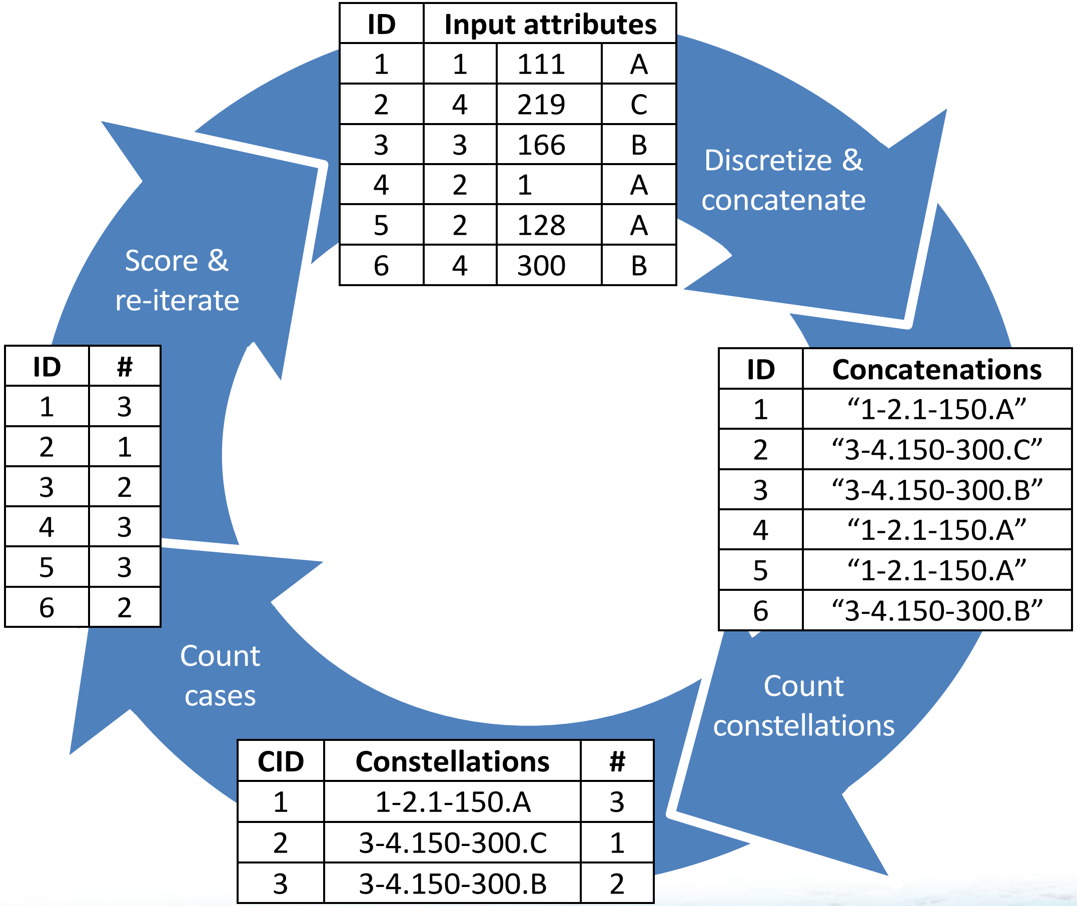
- Largely *density-based* (histogram-based approach).
- Employs *discretization* of continuous attributes to jointly take into account both categorical and numerical variables (concatenation trick).
- Concatenations (combinations) of attributes are referred to as *constellations* when determining their density.
- Works *iteratively* so as to obtain ever narrower discretization intervals, which avoids arbitrary and suboptimal bin sizes.
- Can capture interactions and other *relationships* between variables.
- Uses exponentially increasing *weights* and *arity*.

The concatenation trick

- An operation that facilitates the joint analysis of categorical and continuous (numerical) attributes.
- Combines continuous and nominal attributes into a string value.
- The concatenation trick requires the discretization of the continuous variables, so that they can be combined (concatenated) with the categorical variables into a single string – and subsequently analyzed.
- Recursive discretization can minimize discretization error and, depending on the goal, result in precise numerical analysis.
- Concatenation also allows for capturing relationships between the attributes.

The SECODA process

1. The process starts by *discretizing* the continuous attributes into b = 2 equiwidth bins (i.e. equal interval ranges) in the first iteration.
2. Each case's categorical and discretized numerical values are then *concatenated*, which yields the *constellations* on which the joint density distribution will be based.
3. By subsequently calculating the constellation *frequencies* (density) it can be determined how rare each case is in the current iteration i.
4. Each case g's average anomaly score $aas_{g,i}$ can be calculated in each iteration i by calculating the arithmetic mean of the case's current constellation frequency and its average score of the previous iteration.
5. In the next round this process is *repeated* with a higher number for b.
6. The algorithm *converges* when a given fraction of the cases in the original dataset has a score below the anomaly threshold.
   - The fraction is set to 0.003 (cf. 3 times the standard deviation).
   - The anomaly threshold starts at 1 (i.e. a truly unique case).



ID	Input attributes		
1	1	111	A
2	4	219	C
3	3	166	B
4	2	1	A
5	2	128	A
6	4	300	B

Discretize & concatenate

ID	Concatenations
1	“1-2.1-150.A”
2	“3-4.150-300.C”
3	“3-4.150-300.B”
4	“1-2.1-150.A”
5	“1-2.1-150.A”
6	“3-4.150-300.B”

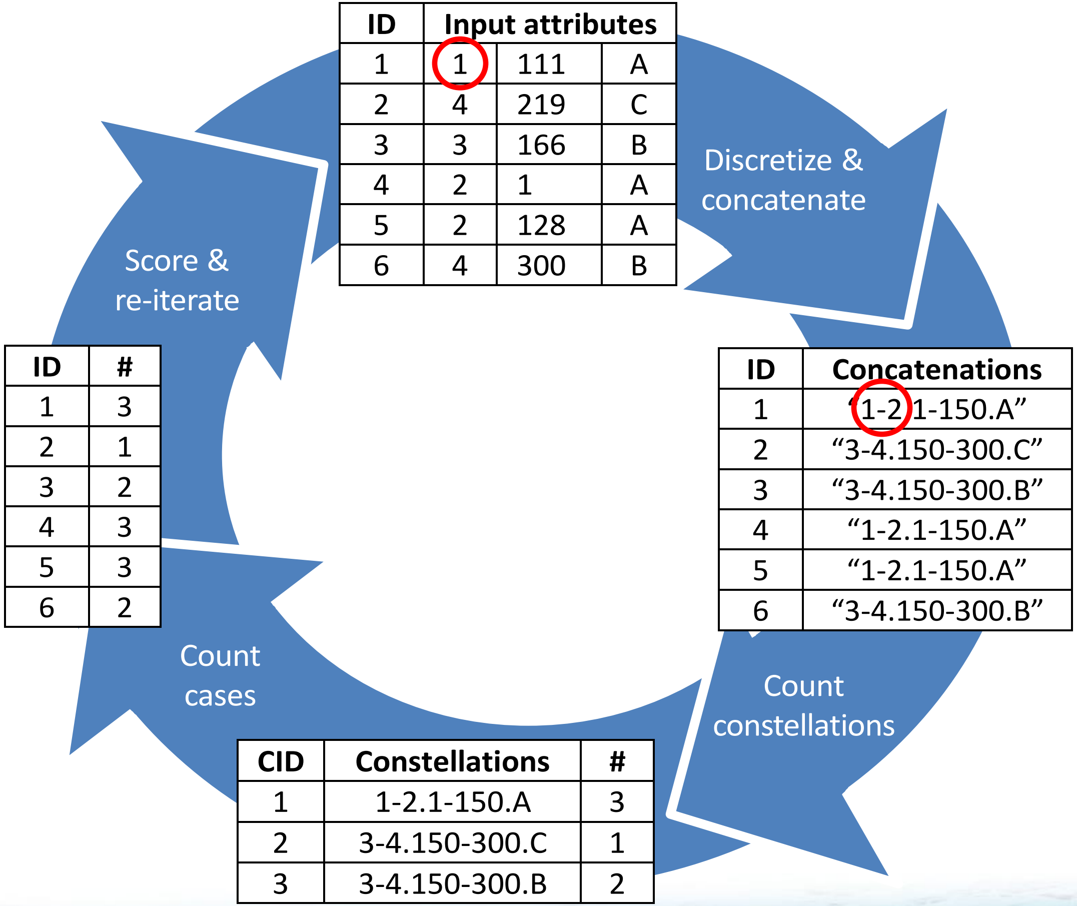
Count constellations

CID	Constellations	#
1	1-2.1-150.A	3
2	3-4.150-300.C	1
3	3-4.150-300.B	2

Count cases

ID	#
1	3
2	1
3	2
4	3
5	3
6	2

Score & re-iterate



ID	Input attributes		
1	1	111	A
2	4	219	C
3	3	166	B
4	2	1	A
5	2	128	A
6	4	300	B

Discretize & concatenate

ID	Concatenations
1	“1-2.1-150.A”
2	“3-4.150-300.C”
3	“3-4.150-300.B”
4	“1-2.1-150.A”
5	“1-2.1-150.A”
6	“3-4.150-300.B”

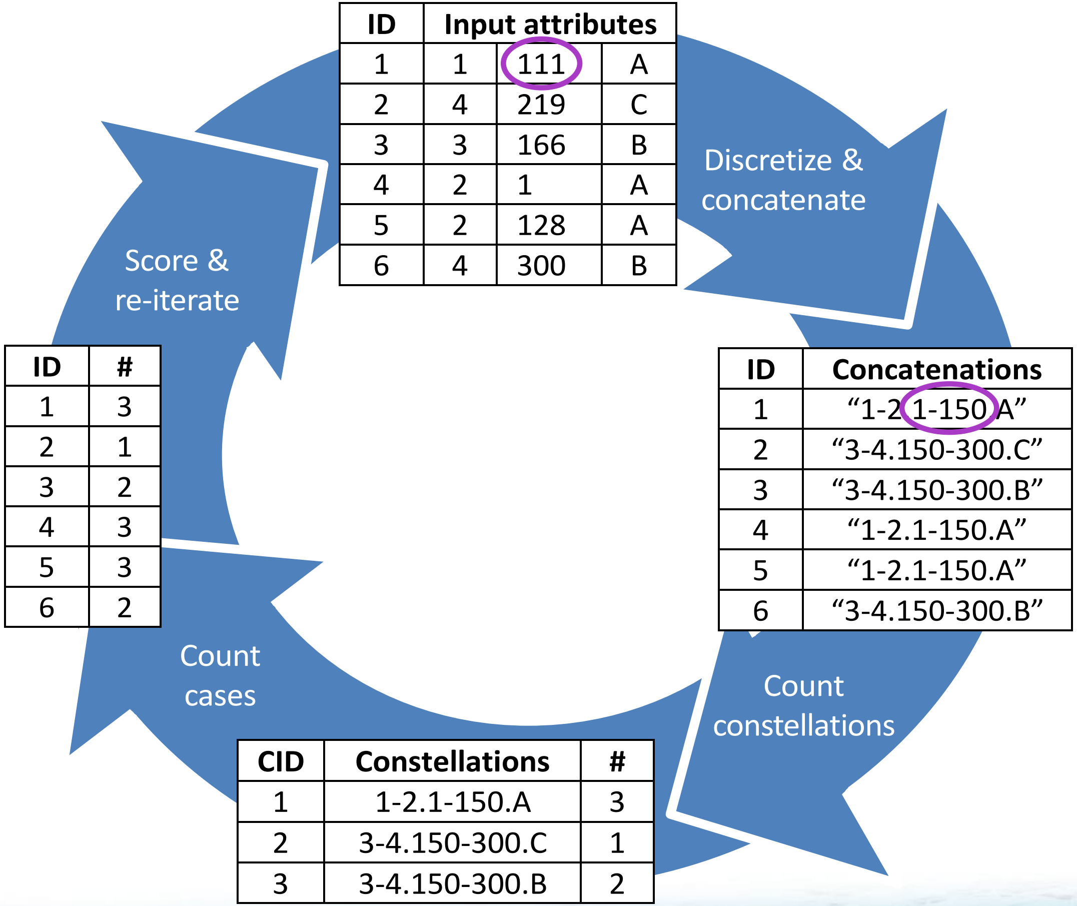
Count constellations

CID	Constellations	#
1	1-2.1-150.A	3
2	3-4.150-300.C	1
3	3-4.150-300.B	2

Count cases

ID	#
1	3
2	1
3	2
4	3
5	3
6	2

Score & re-iterate



ID	Input attributes		
1	1	111	A
2	4	219	C
3	3	166	B
4	2	1	A
5	2	128	A
6	4	300	B

Discretize & concatenate

ID	Concatenations
1	“1-2.1-150.A”
2	“3-4.150-300.C”
3	“3-4.150-300.B”
4	“1-2.1-150.A”
5	“1-2.1-150.A”
6	“3-4.150-300.B”

Count constellations

CID	Constellations	#
1	1-2.1-150.A	3
2	3-4.150-300.C	1
3	3-4.150-300.B	2

Count cases

ID	#
1	3
2	1
3	2
4	3
5	3
6	2

Score & re-iterate

The SECODA process

Tactics for speeding up the analysis:

- *Exponentially increasing weights*: The last iteration weighs as much as the previous iterations combined (i.e. implicit weights). This also prevents bias.
- *Pruning heuristic*: Prunes away that part from the search space that represent the most normal cases.
- *Increased arity*: Increasing the number of discretized bins with larger steps as the analysis process continues.

The SECODA algorithm

See the paper and R code for precise implementation and detailed comments.

URL: www.foorthuis.nl

```
Algorithm: SECODA
Inputs: $\boldsymbol{D_0}$, the original matrix with $n$ cases and $p$ attributes.
Output: $\boldsymbol{aas}_i$ , a vector of average anomaly scores after the last itera-
  tion for all cases in $\boldsymbol{D_0}$, with $aas_{g,i}$ representing the individual score.
Key local vars:   $b$, the number of discretization bins (arity).
                  $s$, used as stop point and for increased binning.
                  $cf_{g,i}$, the current frequency in iteration $i$ of the
                    constellation to which case $g$ belongs.
begin
    $i \leftarrow 0$; $b \leftarrow 2$; $s \leftarrow 1$; $continue \leftarrow$ TRUE  # Set initial values
    while $continue =$ TRUE do
        $i \leftarrow i + 1$
        $\boldsymbol{D'} \leftarrow \boldsymbol{D}_i$ with numerical attributes discretized into $b$ equiwidth bins
        $cf_{g,i} \leftarrow$ ConstellationFrequencyPerCase($\boldsymbol{D'}$)
        if $i > 1$  # Calculate average anomaly scores for cases in $\boldsymbol{D}_i$
          $aas_{g,i} \leftarrow \frac{1}{2}(aas_{g,i-1} + cf_{g,i})$
        else  # If it's the first iteration, put in the frequency
          $aas_{g,i} \leftarrow cf_{g,i}$
        end if
        if $i \leq 10$  # Iteration management
          $s \leftarrow s + 0.1$
          $b \leftarrow b + 1$
        else  # Take larger steps and prune cases in higher iterations
          $s \leftarrow s + 1$
          $b \leftarrow b + (s - 2)$
          # Add to $\boldsymbol{aasp}_i$ the anomaly scores of the 5% most normal cases
            that are to be pruned away:
          $\boldsymbol{p} \leftarrow$ subset of $\boldsymbol{aas}_i$, with each $aas_{g,i} \geq 0.95$ quantile value
          $\boldsymbol{aasp}_i \leftarrow \boldsymbol{aasp}_{i-1} \cup \boldsymbol{p}$
          # Prune away high-frequency (normal) cases for next iteration:
          $\boldsymbol{D}_{i+1} \leftarrow$ subset of $\boldsymbol{D}_i$, with each case such that
            its $aas_{g,i} < 0.95$ quantile value
        end if
        $\boldsymbol{Q} \leftarrow$ Subset of $\boldsymbol{D}_i$, with each case such that its $aas_{g,i} \leq s$
        if (noc($\boldsymbol{Q}$) / noc($\boldsymbol{D_0}$)) > 0.003 # Verify fraction of identified anomalies
          $continue \leftarrow$ FALSE  # No new iteration (process has converged)
        end if
    end while
    $\boldsymbol{aas}_i \leftarrow \boldsymbol{aas}_i \cup \boldsymbol{aasp}_{i-1}$  # Combine average anomaly scores from latest
      iteration with scores from cases that have been pruned previously
    return $\boldsymbol{aas}_i$  # Return full anomaly score vector as the end result
end
```

```
Algorithm: ConstellationFrequencyPerCase
Inputs: $\boldsymbol{D'}$, containing $p$ (categorical and discretized numerical) attributes and a total of $n$ cases, with $n \leq$ noc($\boldsymbol{D_0}$).
Output: $\boldsymbol{cf}_i$ , a vector with for each case $cf_{g,i}$ the frequency of the constellation to which the case belongs in the current iteration.
begin
    # Concatenate each case's attribute values in this iteration (i.e. determine the constellations):
    $cc_{g,i} \leftarrow d'_{g,1,i} \oplus d'_{g,2,i} \oplus \ldots \oplus d'_{g,p,i}$
    # Determine the frequency of distinct constellations in this iteration (with $k$ identifying the constellations):
    $ccf_{k,i} \leftarrow$ The number of cases per constellation
    # Determine the frequency of each case, using the frequencies of their constellations (i.e. inner join $\boldsymbol{cc}_i$ and $\boldsymbol{ccf}_i$ on $k$):
    $cf_{g,i} \leftarrow$ The frequency from $ccf_{k,i}$ for each case's corresponding constellation
    return $\boldsymbol{cf}_i$  # Return each case's current frequency $cf_{g,i}$ as the elements of a vector
end
```

Algorithm evaluation

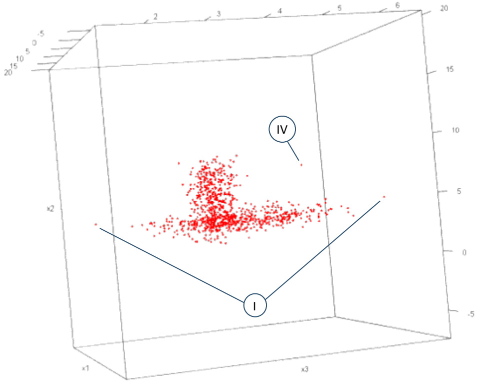
Three types of evaluations:

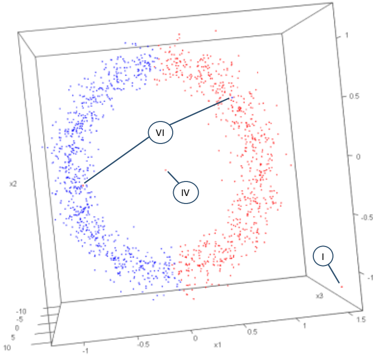
1. Simulated datasets are used to study whether SECODA is capable of *identifying* the different types of anomalies.
2. Two real-world datasets with labeled anomalies (test sets) are used to evaluate SECODA with ROC & PRC curves and related *performance* metrics.
3. The results of a real-world *data quality use case* are presented.

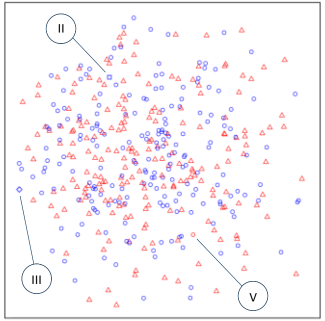
Experiments were conducted in R 3.3.2.

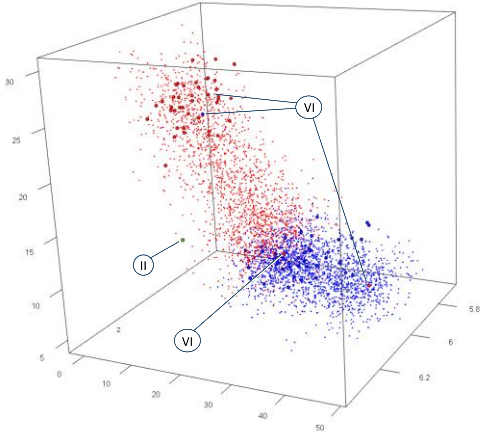
1. Simulations

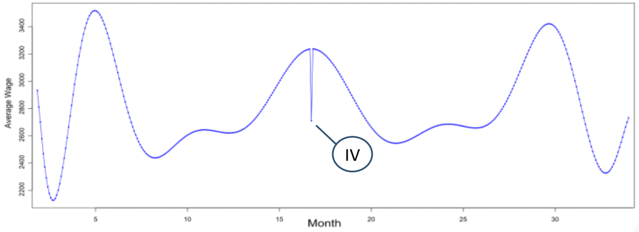
All six anomaly types were identified in the simulated datasets.





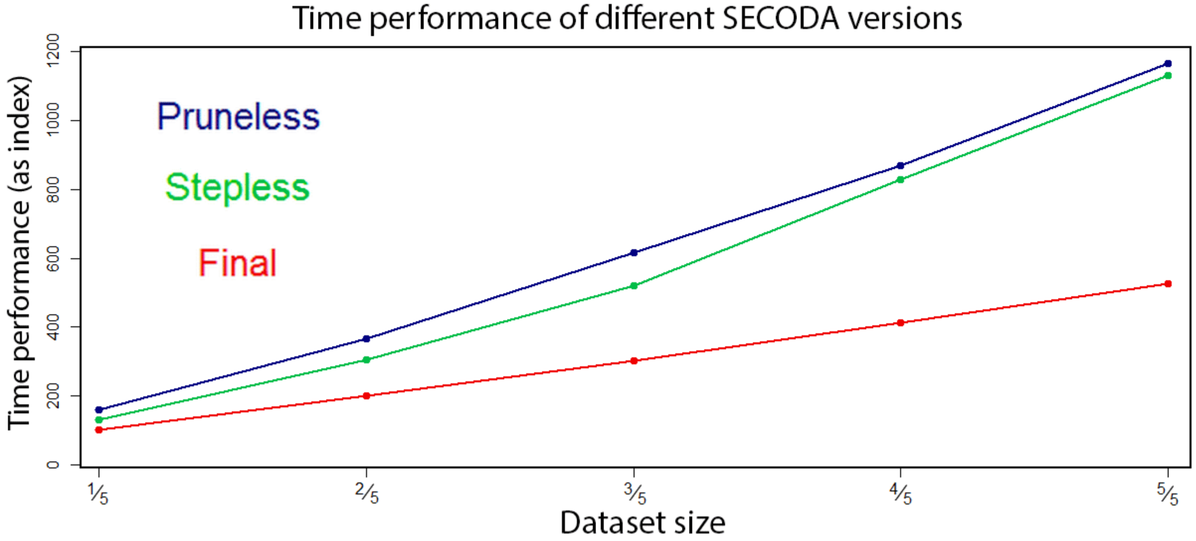






2. Test sets

Time performance scales linearly with dataset size.



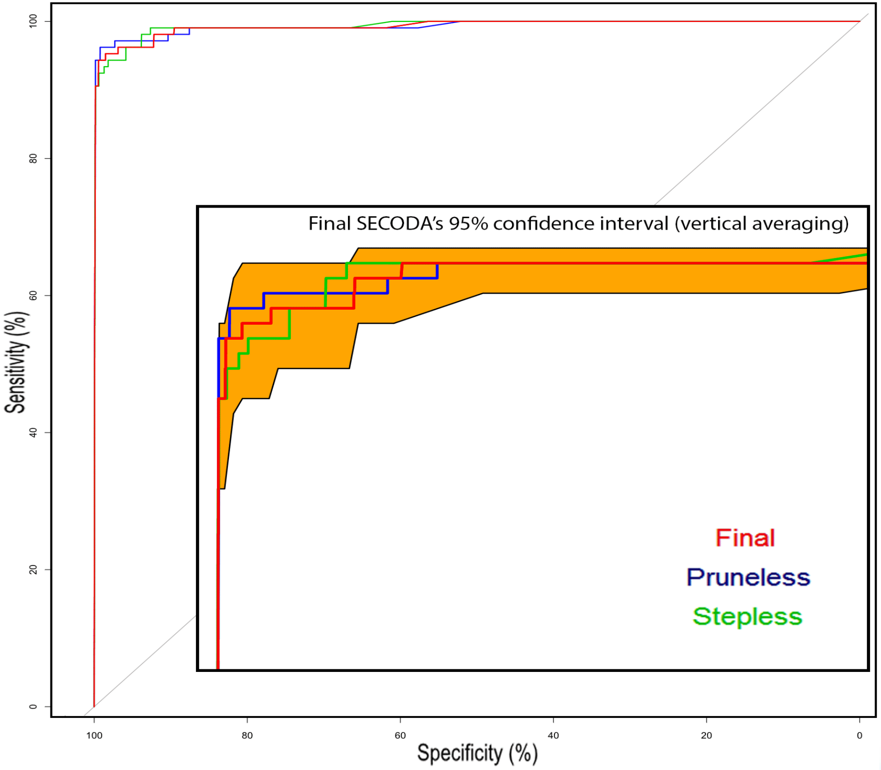
Thus the heuristics are effective in speeding up the analysis.

But: does the final algorithm with the heuristics perform as well as the others?

2. Test sets

Functionality:
ROC & PRC

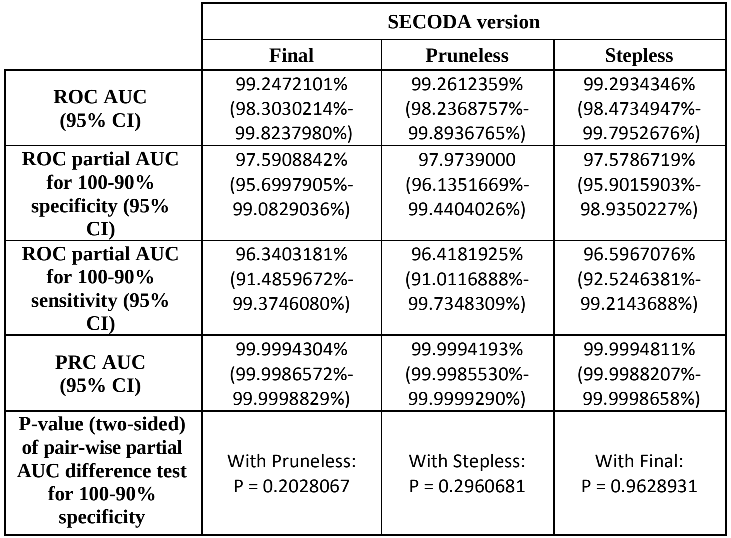
For testing the performance of the different versions of the algorithm



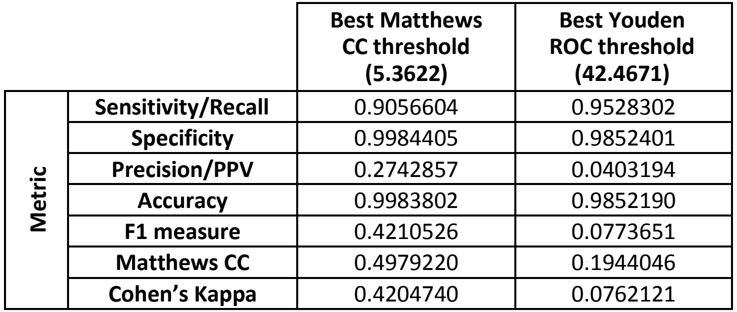
2. Test sets

Functionality:

Metrics



	SECODA version		
	Final	**Pruneless**	**Stepless**
ROC AUC (95% CI)	99.2472101% (98.3030214%-99.8237980%)	99.2612359% (98.2368757%-99.8936765%)	99.2934346% (98.4734947%-99.7952676%)
ROC partial AUC for 100-90% specificity (95% CI)	97.5908842% (95.6997905%-99.0829036%)	97.9739000 (96.1351669%-99.4404026%)	97.5786719% (95.9015903%-98.9350227%)
ROC partial AUC for 100-90% sensitivity (95% CI)	96.3403181% (91.4859672%-99.3746080%)	96.4181925% (91.0116888%-99.7348309%)	96.5967076% (92.5246381%-99.2143688%)
PRC AUC (95% CI)	99.9994304% (99.9986572%-99.9998829%)	99.9994193% (99.9985530%-99.9999290%)	99.9994811% (99.9988207%-99.9998658%)
P-value (two-sided) of pair-wise partial AUC difference test for 100-90% specificity	With Pruneless: P = 0.2028067	With Stepless: P = 0.2960681	With Final: P = 0.9628931

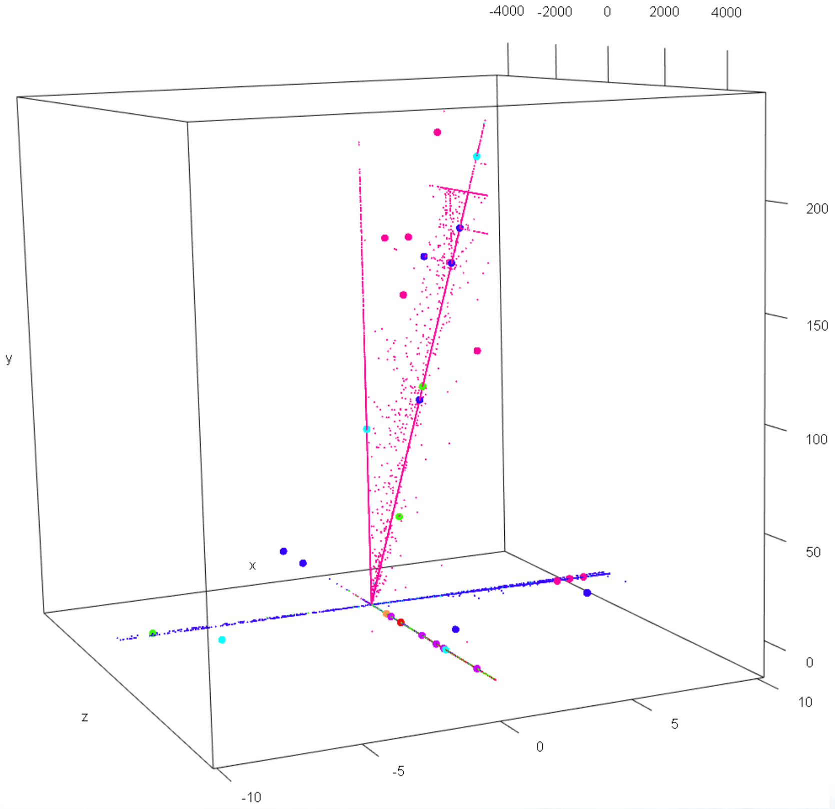


		Best Matthews CC threshold (5.3622)	**Best Youden ROC threshold (42.4671)**
Metric	**Sensitivity/Recall**	0.9056604	0.9528302
	Specificity	0.9984405	0.9852401
	Precision/PPV	0.2742857	0.0403194
	Accuracy	0.9983802	0.9852190
	F1 measure	0.4210526	0.0773651
	Matthews CC	0.4979220	0.1944046
	Cohen’s Kappa	0.4204740	0.0762121

Confidence intervals have been calculated with 10000 stratified percentile bootstrap resamples.

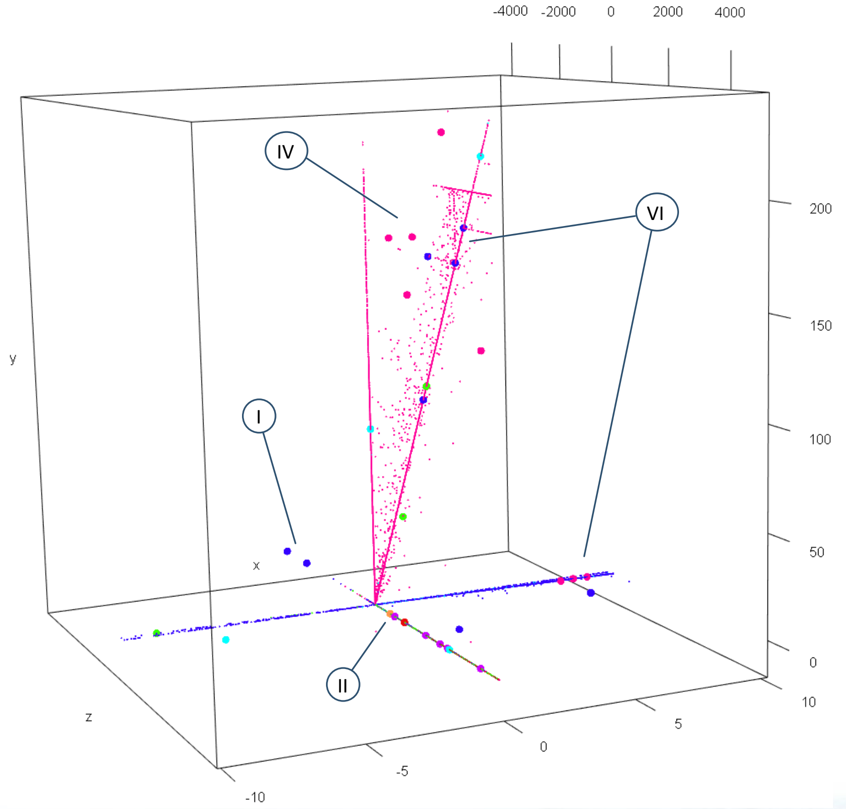
No significant differences exist between the individual curves, nor between the AUCs. Thus: the different heuristics used to boost the time performance of SECODA do not have an adverse effect on its functional ability to detect true anomalies.

3. Real-world Polis Administration case



3. Real-world Polis Administration case

Different types of anomalies were detected.



Discussion

- No need for SECODA to calculate point-to-point distances or associations.
- Low memory requirements.
- Can deal with complex relationships between variables.
- The concatenation trick facilitates the analysis of mixed data and missing values, and is not affected by multicollinearity.
- Exponentially increasing weights: This speeds up the analysis, prevents bias and results in a low memory imprint.
- The pruning heuristic is a self-regulating mechanism during runtime and dynamically decides how many cases to discard.
- Affords parallel processing.
- Alas, the curse of dimensionality still holds (depends on the amount of attributes, cases, classes, distribution, etc).

Discussion

SECODA bears similarities with:

- Density-based algorithms. Mainly histogram-based, as used in e.g. intrusion detection systems [6, 22, 33, 34, 50, 51].
- Ensembles, especially iForest [28, 29].
- The high-dimensional outlier detection presented in [6].

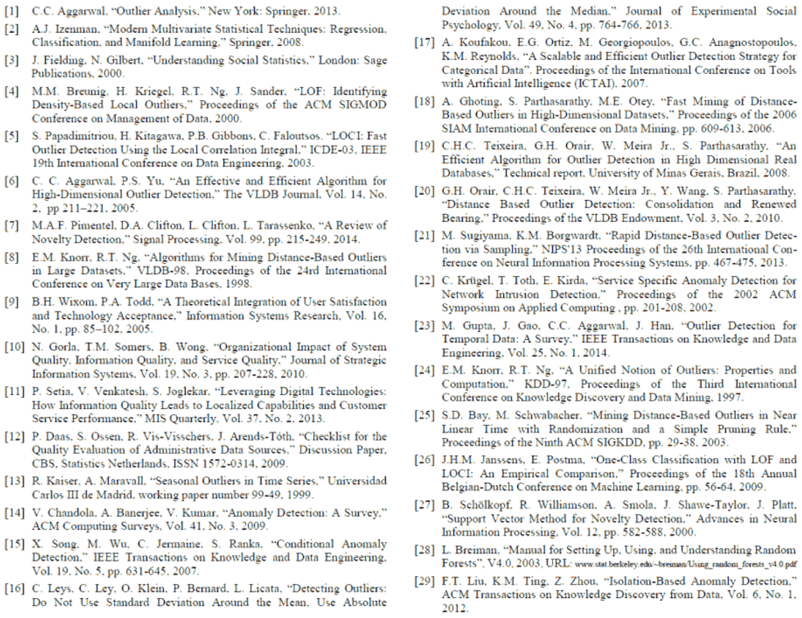
SECODA and AD can contribute to several well-known data quality aspects [cf. 12, 44, 45].

- Correctness of individual values
- Completeness of cases
- Consistency between attribute values

Contributions

- Introduction of a *typology of anomalies,* which can be used for understanding the different anomaly types in datasets and the evaluation of AD algorithms.
- New general-purpose *algorithm* for the detection of multiple types of anomalies.
- Shown that complex anomalies can be identified by a relatively *simple* algorithm using basic data operations (without point-to-point calculations or complex fitting procedures). This allows in-database analytics, parallel processing and the analysis of very large datasets.
- The real-world case shows that SECODA, and AD in general, can be used in practice to improve *data quality*.

References



[1] C.C. Aggarwal, “Outlier Analysis,” New York: Springer, 2013.

[2] A.J. Izenman, “Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning,” Springer, 2008.

[3] J. Fielding, N. Gilbert, “Understanding Social Statistics,” London: Sage Publications, 2000.

[4] M.M. Breunig, H. Kriegel, R.T. Ng, J. Sander, “LOF: Identifying Density-Based Local Outliers,” Proceedings of the ACM SIGMOD Conference on Management of Data, 2000.

[5] S. Papadimitriou, H. Kitagawa, P.B. Gibbons, C. Faloutsos, “LOCI: Fast Outlier Detection Using the Local Correlation Integral,” ICDE-03, IEEE 19th International Conference on Data Engineering, 2003.

[6] C. C. Aggarwal, P.S. Yu, “An Effective and Efficient Algorithm for High-Dimensional Outlier Detection,” The VLDB Journal, Vol. 14, No. 2, pp 211–221, 2005.

[7] M.A.F. Pimentel, D.A. Clifton, L. Clifton, L. Tarassenko, “A Review of Novelty Detection,” Signal Processing, Vol. 99, pp. 215-249, 2014.

[8] E.M. Knorr, R.T. Ng, “Algorithms for Mining Distance-Based Outliers in Large Datasets,” VLDB-98, Proceedings of the 24rd International Conference on Very Large Data Bases, 1998.

[9] B.H. Wixom, P.A. Todd, “A Theoretical Integration of User Satisfaction and Technology Acceptance,” Information Systems Research, Vol. 16, No. 1, pp. 85–102, 2005.

[10] N. Gorla, T.M. Somers, B. Wong, “Organizational Impact of System Quality, Information Quality, and Service Quality,” Journal of Strategic Information Systems, Vol. 19, No. 3, pp. 207-228, 2010.

[11] P. Setia, V. Venkatesh, S. Joglekar, “Leveraging Digital Technologies: How Information Quality Leads to Localized Capabilities and Customer Service Performance,” MIS Quarterly, Vol. 37, No. 2, 2013.

[12] P. Daas, S. Ossen, R. Vis-Visschers, J. Arends-Tóth, “Checklist for the Quality Evaluation of Administrative Data Sources,” Discussion Paper, CBS, Statistics Netherlands, ISSN 1572-0314, 2009.

[13] R. Kaiser, A. Maravall, “Seasonal Outliers in Time Series,” Universidad Carlos III de Madrid, working paper number 99-49, 1999.

[14] V. Chandola, A. Banerjee, V. Kumar, “Anomaly Detection: A Survey,” ACM Computing Surveys, Vol. 41, No. 3, 2009.

[15] X. Song, M. Wu, C. Jermaine, S. Ranka, “Conditional Anomaly Detection,” IEEE Transactions on Knowledge and Data Engineering, Vol. 19, No. 5, pp. 631-645, 2007.

[16] C. Leys, C. Ley, O. Klein, P. Bernard, L. Licata, “Detecting Outliers: Do Not Use Standard Deviation Around the Mean, Use Absolute Deviation Around the Median,” Journal of Experimental Social Psychology, Vol. 49, No. 4, pp. 764-766, 2013.

[17] A. Koufakou, E.G. Ortiz, M. Georgiopoulos, G.C. Anagnostopoulos, K.M. Reynolds, “A Scalable and Efficient Outlier Detection Strategy for Categorical Data”. Proceedings of the International Conference on Tools with Artificial Intelligence (ICTAI), 2007.

[18] A. Ghoting, S. Parthasarathy, M.E. Otey, “Fast Mining of Distance-Based Outliers in High-Dimensional Datasets,” Proceedings of the 2006 SIAM International Conference on Data Mining, pp. 609-613, 2006.

[19] C.H.C. Teixeira, G.H. Orair, W. Meira Jr., S. Parthasarathy, “An Efficient Algorithm for Outlier Detection in High Dimensional Real Databases,” Technical report, University of Minas Gerais, Brazil, 2008.

[20] G.H. Orair, C.H.C. Teixeira, W. Meira Jr., Y. Wang, S. Parthasarathy, “Distance Based Outlier Detection: Consolidation and Renewed Bearing,” Proceedings of the VLDB Endowment, Vol. 3, No. 2, 2010.

[21] M. Sugiyama, K.M. Borgwardt, “Rapid Distance-Based Outlier Detection via Sampling,” NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems, pp. 467-475, 2013.

[22] C. Krügel, T. Toth, E. Kirda, “Service Specific Anomaly Detection for Network Intrusion Detection,” Proceedings of the 2002 ACM Symposium on Applied Computing , pp. 201-208, 2002.

[23] M. Gupta, J. Gao, C.C. Aggarwal, J. Han, “Outlier Detection for Temporal Data: A Survey,” IEEE Transactions on Knowledge and Data Engineering, Vol. 25, No. 1, 2014.

[24] E.M. Knorr, R.T. Ng, “A Unified Notion of Outliers: Properties and Computation,” KDD-97, Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, 1997.

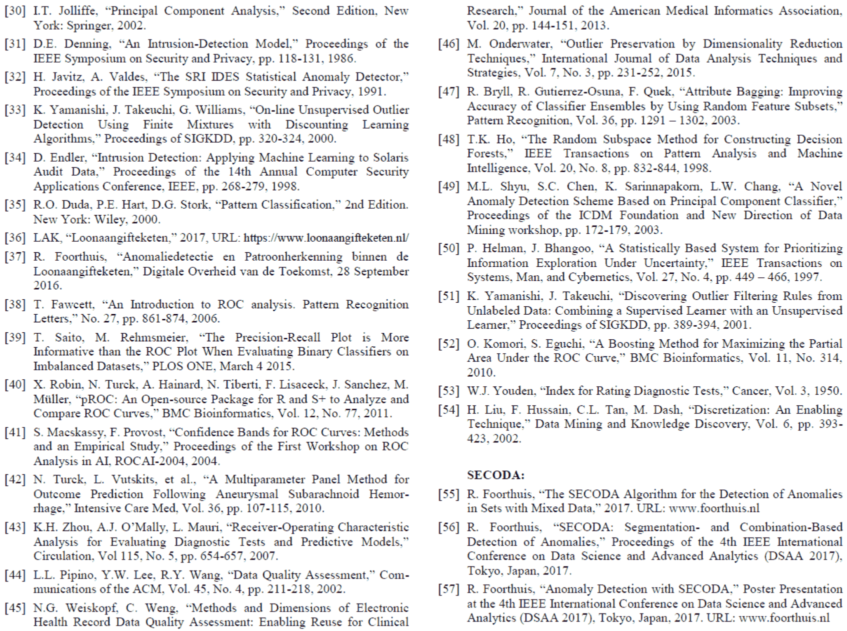
[25] S.D. Bay, M. Schwabacher, “Mining Distance-Based Outliers in Near Linear Time with Randomization and a Simple Pruning Rule,” Proceedings of the Ninth ACM SIGKDD, pp. 29-38, 2003.

[26] J.H.M. Janssens, E. Postma, “One-Class Classification with LOF and LOCI: An Empirical Comparison,” Proceedings of the 18th Annual Belgian-Dutch Conference on Machine Learning, pp. 56-64, 2009.

[27] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, J. Platt, “Support Vector Method for Novelty Detection,” Advances in Neural Information Processing, Vol. 12, pp. 582-588, 2000.

[28] L. Breiman, “Manual for Setting Up, Using, and Understanding Random Forests”, V4.0, 2003, URL: www.stat.berkeley.edu/~breiman/Using_random_forests_v4.0.pdf

[29] F.T. Liu, K.M. Ting, Z. Zhou, “Isolation-Based Anomaly Detection,” ACM Transactions on Knowledge Discovery from Data, Vol. 6, No. 1, 2012.



[30] I.T. Jolliffe, “Principal Component Analysis,” Second Edition, New York: Springer, 2002.

[31] D.E. Denning, “An Intrusion-Detection Model,” Proceedings of the IEEE Symposium on Security and Privacy, pp. 118-131, 1986.

[32] H. Javitz, A. Valdes, “The SRI IDES Statistical Anomaly Detector,” Proceedings of the IEEE Symposium on Security and Privacy, 1991.

[33] K. Yamanishi, J. Takeuchi, G. Williams, “On-line Unsupervised Outlier Detection Using Finite Mixtures with Discounting Learning Algorithms,” Proceedings of SIGKDD, pp. 320-324, 2000.

[34] D. Endler, “Intrusion Detection: Applying Machine Learning to Solaris Audit Data,” Proceedings of the 14th Annual Computer Security Applications Conference, IEEE, pp. 268-279, 1998.

[35] R.O. Duda, P.E. Hart, D.G. Stork, “Pattern Classification,” 2nd Edition, New York: Wiley, 2000.

[36] LAK, “Loonaangifteketen,” 2017, URL: https://www.loonaangifteketen.nl/

[37] R. Foorthuis, “Anomaliedetectie en Patroonherkenning binnen de Loonaangifteketen,” Digitale Overheid van de Toekomst, 28 September 2016.

[38] T. Fawcett, “An Introduction to ROC analysis. Pattern Recognition Letters,” No. 27, pp. 861-874, 2006.

[39] T. Saito, M. Rehmsmeier, “The Precision-Recall Plot is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets,” PLOS ONE, March 4 2015.

[40] X. Robin, N. Turck, A. Hainard, N. Tiberti, F. Lisaceck, J. Sanchez, M. Müller, “pROC: An Open-source Package for R and S+ to Analyze and Compare ROC Curves,” BMC Bioinformatics, Vol. 12, No. 77, 2011.

[41] S. Macskassy, F. Provost, “Confidence Bands for ROC Curves: Methods and an Empirical Study,” Proceedings of the First Workshop on ROC Analysis in AI, ROCAI-2004, 2004.

[42] N. Turck, L. Vutskits, et al., “A Multiparameter Panel Method for Outcome Prediction Following Aneurysmal Subarachnoid Hemorrhage,” Intensive Care Med, Vol. 36, pp. 107-115, 2010.

[43] K.H. Zhou, A.J. O'Mally, L. Mauri, “Receiver-Operating Characteristic Analysis for Evaluating Diagnostic Tests and Predictive Models,” Circulation, Vol 115, No. 5, pp. 654-657, 2007.

[44] L.L. Pipino, Y.W. Lee, R.Y. Wang, “Data Quality Assessment,” Communications of the ACM, Vol. 45, No. 4, pp. 211-218, 2002.

[45] N.G. Weiskopf, C. Weng, “Methods and Dimensions of Electronic Health Record Data Quality Assessment: Enabling Reuse for Clinical Research,” Journal of the American Medical Informatics Association, Vol. 20, pp. 144-151, 2013.

[46] M. Onderwater, “Outlier Preservation by Dimensionality Reduction Techniques,” International Journal of Data Analysis Techniques and Strategies, Vol. 7, No. 3, pp. 231-252, 2015.

[47] R. Bryll, R. Gutierrez-Osuna, F. Quek, “Attribute Bagging: Improving Accuracy of Classifier Ensembles by Using Random Feature Subsets,” Pattern Recognition, Vol. 36, pp. 1291 – 1302, 2003.

[48] T.K. Ho, “The Random Subspace Method for Constructing Decision Forests,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 8, pp. 832-844, 1998.

[49] M.L. Shyu, S.C. Chen, K. Sarinnapakorn, L.W. Chang, “A Novel Anomaly Detection Scheme Based on Principal Component Classifier,” Proceedings of the ICDM Foundation and New Direction of Data Mining workshop, pp. 172-179, 2003.

[50] P. Helman, J. Bhangoo, “A Statistically Based System for Prioritizing Information Exploration Under Uncertainty,” IEEE Transactions on Systems, Man, and Cybernetics, Vol. 27, No. 4, pp. 449 – 466, 1997.

[51] K. Yamanishi, J. Takeuchi, “Discovering Outlier Filtering Rules from Unlabeled Data: Combining a Supervised Learner with an Unsupervised Learner,” Proceedings of SIGKDD, pp. 389-394, 2001.

[52] O. Komori, S. Eguchi, “A Boosting Method for Maximizing the Partial Area Under the ROC Curve,” BMC Bioinformatics, Vol. 11, No. 314, 2010.

[53] W.J. Youden, “Index for Rating Diagnostic Tests,” Cancer, Vol. 3, 1950.

[54] H. Liu, F. Hussain, C.L. Tan, M. Dash, “Discretization: An Enabling Technique,” Data Mining and Knowledge Discovery, Vol. 6, pp. 393-423, 2002.

SECODA:

[55] R. Foorthuis, “The SECODA Algorithm for the Detection of Anomalies in Sets with Mixed Data,” 2017. URL: www.foorthuis.nl

[56] R. Foorthuis, “SECODA: Segmentation- and Combination-Based Detection of Anomalies,” Proceedings of the 4th IEEE International Conference on Data Science and Advanced Analytics (DSAA 2017), Tokyo, Japan, 2017.

[57] R. Foorthuis, “Anomaly Detection with SECODA,” Poster Presentation at the 4th IEEE International Conference on Data Science and Advanced Analytics (DSAA 2017), Tokyo, Japan, 2017. URL: www.foorthuis.nl