Transactions on Machine Learning
and Data Mining
Vol. 9, No.2 (2016) 48-61
© ISSN: 1865-6781 (Journal)
ISBN: 978-3-942952-47-7
IBaI-Publishing ISSN 1864-9734

ibai Publishing
www.ibai-publishing.org

Preprocessing Optimization for Predictive Classification: Baseline Results from Six Industry Cases

Markus Vattulainen

School of Information Sciences, University of Tampere, Finland

markus.vattulainen@gmail.com

Abstract. Data preprocessing is often the most time-consuming phase in data analysis and automation of it requires computationally costly search from preprocessing combinations. Efforts to build and evaluate efficient preprocessing automation systems have been challenged by the lack of baseline results from industry regarding the extent of which the infeasible exhaustive search can be speeded up. The research question addressed is: how good are heuristic search methods compared to exhaustive search given a 10%-time constraint? The baseline results from 5/6 real business performance measurement system cases show that simple hill-climbing heuristic with one or three restarts resulted in median 98% classification accuracy compared to global optimum found by exhaustive search. The outcome is attributed to the characteristics of the search space, which included several points near the optimum in all of the cases. For the worst case heuristic hyperparameter optimization with hybridization increased the comparative ratio from 82% to 89%. The results suggest that faster heuristic methods can find near-optimal preprocessing combinations and thus support efficient automation of preprocessing for predictive classification.

Keywords: Preprocessing, Classification, Optimization, Metaheuristics, Business performance measurement system

1 Introduction

Business performance measurement system is an important tool in the implementation of strategy. Current business performance measurement systems aim to balance financial measures with non-financial ones to ensure long-term share-holder value creation

and to provide predictive power not only reporting what has happened but also what will happen [20]. The aims are supported by the processes of metrics design, data gathering and manipulation, and data analysis [13]. Of the latter two, data gathering and manipulation (i.e. preprocessing) is often manual work and the most time-consuming phase [29]. Exploration of what can be efficiently automated is a fundamental computer science research objective [10]. Preprocessing automation objective in the business performance measurement system context is to find and execute a combination of preprocessing techniques that maximizes a predictive classification performance metric such as classification accuracy. Automation should be efficient and scalable.

There are no known analytical solutions to the problem of finding the best preprocessing combination nor is it feasible to do exhaustive evaluation of all combinations. A gap in the literature of preprocessing automation design is the lack of baseline results from real industry cases regarding performance of heuristic methods. Thus the research question addressed is: how good are heuristic methods compared to exhaustive search given a 10%-time constraint? There are two main limitations to the research question. First, the focus is exclusively on preprocessing and baseline results. The impact of predictive model selection or model hyperparameter tuning is not discussed. No claims are made of best results or novelty in heuristic methods. Therefore, method comparisons are not included. Secondly, the specific 10%-time constraint is only weakly motivated. It was set as it is by running various experiments with the cases. The practical objective was to limit the search time used for each case to a single working day.

A metaheuristic optimization framework for preprocessing was built and six cases from the business performance measurement system domain were acquired to compute the baseline results: Toyota Material Handling Finland, 3StepIT, M-Files, Innolink, Papua and Lempesti. The data sets had one financial target feature (e.g. customer profit margin, sales person sales volume), 7 to 46 non-financial numerical predictors and the number of data points varied from 48 to 344. The measured data objects were employees, customers and process runs. The baseline results are: in 5/6 cases simple hill-climbing with one or three restarts reached median 98% level in classification accuracy compared to global optimum in 10% of time compared to exhaustive search. This paper contributes to the body of existing knowledge on preprocessing system design by: characterizing the search space of preprocessing combinations for classification presenting baseline results from six real industry cases. The results suggest that preprocessing can be efficiently automated without significant loss of solution quality.

2 Related research

The related research (Table 1.) can be categorized to consist of preprocessing foundations and preprocessing automation. The latter can be achieved either by optimization or learned policy. The need of preprocessing originates from data quality issues in the information system design [38], poor data quality practices [32] and requirements of the data analysis methods themselves. Pyle [29] estimates that preprocessing can take up to 85% of an analysis project. On a high level preprocessing is understood to consist of data cleaning, data integration, data reduction and data transformation [17] and is a part of data mining standards like CRISP-DM (see [8]). There are preprocessing phases such as low variance removal, value range scaling, noise smoothing, outlier detection,

missing value imputation, class imbalance correction, duplicate detection and removal of irrelevant features. Each phase has a set of competing techniques such as oversampling, undersampling and SMOTE [25] for class imbalance problem. Also, there

Table 1. Related research

Category	Topic	Maturity
1. Foundations	need of preprocessing	High
	techniques	High
	phases	High
	standards	High
2. Automation	combinations	Medium
2a. Optimization	optimization frameworks	High
	baseline results	Low
2b. Policies	reinforcement learning	Low
	Monte Carlo tree search	Low

are application domain motivated preprocessing instructions and studies e.g. for multimedia data [28], direct marketing data [9] and chemometric data [11].

Several articles [12, 40, 22, 39] on the state and future of knowledge discovery research acknowledge preprocessing automation as a priority objective. Automation is based on preprocessing combinations. A preprocessing combination can be defined as an ordered (by phase) set of preprocessing techniques. Preprocessing combination studies [9, 11, 35] demonstrate that preprocessing combinations can have significant and unexpected interaction effects. Literature did not show analytic solutions to the preprocessing combinations optimization problem and metaheuristics was selected as an approach over random and grid searches. Glover and Kochenberger [15] define metaheuristics as: “Iterative process that guides the operation of one or more subordinate heuristics (which may be from a local search process to a constructive process of random solutions) to efficiently produce quality solution for a problem”. In the metaheuristic research community current surveys [27, 2, 1] highlight the two transitions in the field: first, as there are no universally best methods and the methods are maturing there is more need for domain-specific problem-method matches instead of or in addition to generic method development. Secondly, hybridization of metaheuristics (i.e. mixing of metaheuristic techniques with each other or with other optimization techniques) [3, 4, 30] is expected to improve performance levels. Parejo et al. [27] conducted an extensive survey of 33 generic metaheuristics optimization frameworks and found gaps in

hyper heuristics, parallel and distributed computing, software engineering best practices and support for hybridization.

On the level of software, optimization of preprocessing combinations has been adressed as part of learning model hyperparameter optimization in Python Hyperopt-Sklearn [24] and partially in R package Caret [23]. However, there are currently no baseline results available regarding the performance of heuristic search methods in the preprocessing of business performance measurement system data. Lastly, as an alternative to optimization policy-based approaches [33, 6] to preprocessing were not found. Policy-based approaches are identified as an important further research opportunity below specifically for their efficiency and scalability.

3 Search Space Characteristics

The preprocessing combinations search space consists of single preprocessing techniques combined by the preprocessing phase they belong to. A random example of a preprocessing combination could be removal of near zero variance variables [23], imputation of missing values by mean impute [18], removal of outliers by their local density LOF [5], correcting class imbalance by synthetic instances [25] and selecting relevant features by using decision trees [16]. This kind of combinations form the search space as illustrated in Table 2. for Innolink case (3/3200 combinations shown).

Table 2. Example of preprocessing combinations, Innolink case.

Nro	Imputation	Variance	Smoothing	Scaling	Outliers	Sampling	Selection
675	Random forest impute	No action	Coarse smooth	Decimal scale	LOF outlier	Over-sample	No action
676	Missing value omit	Near-zerovar	Coarse smooth	Decimal scale	LOF outlier	Over-sample	No action
677	Mean impute	Near-zerovar	Coarse smooth	Decimal scale	LOF outlier	Over-sample	No action

The behaviour of the objective function in the preprocessing search space can be characterized by plotting K-nearest neighbour classification accuracies by combination (Figure 1.). It can be observed that the objective function is discontinuous. There are regimes (Innolink case) and patterns (Papua and Lempesti cases). These characteristics suggest that metaheuristics must have strong exploration capability and restart may benefit from evenly-spaced start points.

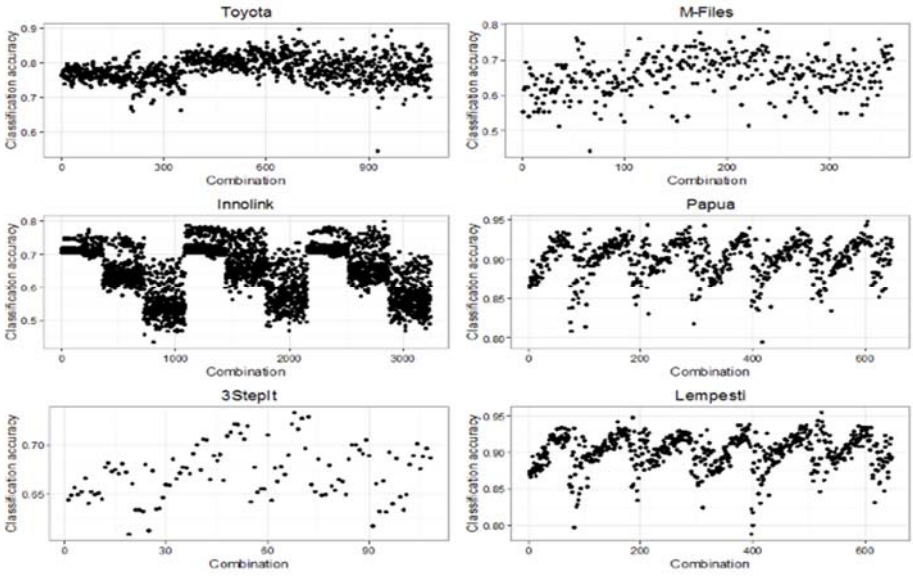

Figure 1. K-nearest neighbour classification accuracies by preprocessing combination

The difference between highest and lowest classification accuracies is substantial supporting the earlier observation that preprocessing can be ineffective [11]. Most importantly, there are several points in the proximity of the global optimum suggesting that heuristics may perform well. Exhaustive search from all combinations is computationally costly. The search space size (Figure 1.) varies between 108 and 3200 combinations. In the cases classification accuracy of a combination was validated by 50 times repeated holdout validation totalling at maximum 160 000 model fitting and prediction events (3200*50) even for a single classifier. The largest case took more than 60 hours on an Intel 1.6 GHz computer. Adding to the cost of holdout validation is the fact that computation of classification accuracies can fail. Failure risks introduced by preprocessing include low or missing variance, presence of missing or infinite values, low observation to variable ratio, class imbalance etc.

4 Method

To provide baseline results a metaheuristic optimization framework for preprocessing was designed, implemented and provided freely accessible. The R [31] package 'metaheur' as well as classes, methods and interface details are presented in [36]. Hybridization capabilities were selected as a design objective due to expected superior perfor-

mance, gap in current systems [27] and hybridization has also the added benefit of focusing design on fundamental building blocks of metaheuristics instead of pseudo innovations as critics have pointed out [34].

4.1 Unified hybrid heuristic

Single-state heuristics included in the study were Hill-Climbing [26], Hill-Climbing with restarts [26], Late-Acceptance Hill-Climbing [7], Taboo search [14], Simulated Annealing [21] and Adaptive simulated annealing [19]. Population-based metaheuristics were not included. For the purpose of hybridization, the essence of each single-state heuristics above was abstracted as a parameter of a unified hybrid heuristic (Table 3).

Table 3. Unified hybrid heuristic

Heuristic	Parameter	SA	Hybrid
Hill-Climbing with restarts	Number of restarts	1	3
Late-Acceptance Hill-Climbing	Previous solution the candidate is compared to	1	2
Taboo search	Number of previous candidates on a taboo list	0	5
Simulated annealing	Probability of accepting an inferior solution (temperature)	X	X
Adaptive simulated Annealing	Probability of increasing the temperature	0	0.1

Each iteration-modification-assessment-selection round [26] consists of a unified heuristic that has all of the above mentioned parameters. The third column in the Table 3. shows an example of using the hybrid to run pure simulated annealing. The fourth column shows an example of a hybrid that consists of three restarts, candidate compared to the solution earlier than current solution, taboo list of length five, simulated annealing temperature and 10% chance of increasing the temperature.

4.2 System components

The hybridization concept above was used to build a metaheuristic optimization framework. The components (Table 4.) are described as follows: Iteration component controls the start and termination conditions. Start class includes start type (single random start with uniform probability, multiple grid restarts or custom start). Custom start (the user specifies the combination number iterations start from) allows the insertion of do-

main knowledge to search. Custom start can be either assumption of the best combination or a combination that has minimum computational complexity. Termination class has the termination conditions: converge, objective threshold or simply the number of iterations run.

Table 4. System components

Component	**Classes**	**Data members**
Iteration	Termination	Termination type
		Number of iterations
		Convergence value
		Threshold value
	Start	Start type Number of starts Start locations
Modification	Tweak	Number of phases tweaked Degree of change (not implemented)
	Taboo	Length of taboo list
Assessment	Objective function	Metric type (default: classification accuracy) Classifier
	Constraint	Combination penalty (not implemented)
Selection	Acceptance	Initial temperature Temperature decrease constant Reheating probability
	Comparison	Location in history candidate is compared to
Control	Hyperparameter opt.	Grid of hyperparameter combinations Random hyperparameters
	Parallelization	Number of cores
	Plotting	Type of plot
	Monitoring	Verbose (true/ false) Logging (true/ false)
	Design of Experiment	Used (true/false)

Table 5. Example of monitoring, Toyota case

[1] "Start type: random restarts."
Number of restarts: 1
Start combination: 941
Iteration: 1 Current best: naomit nearzerovar coarsesmooth minmaxscale noaction noaction 0.615384
Iteration: 1 Candidate: naomit nearzerovar coarsesmooth noaction noaction noaction 0.6923077
Temperature: 0.85
Comparison value for late acceptance: 0.6153846
History delta, last three: 0.07692308
Iteration: 2 Current best: naomit nearzerovar coarsesmooth noaction noaction noaction 0.6923077
Iteration: 2 Candidate: naomit noaction coarsesmooth noaction noaction noaction 0.6923077
Temperature: 0.7225
Comparison value for late acceptance: 0.6923077
SA: A weaker solution was accepted.
History delta, last three: 0.03846154

Modification component is responsible for creating the candidates. Tweak class takes the current best combination and modifies it according to Gaussian convolution [26], i.e. most changes are small but occasional large changes can happen. Taboo class controls the length of list of previously visited candidates that cannot be revisited.

Assessment component computes the classification accuracy for each candidate combination. Objective function class includes classification performance metrics such as accuracy or kappa. Constraints can be defined as a penalty for a combination.

Selection component makes a selection. Acceptance probability class controls the probability of accepting an inferior solution (in order to escape local maximum) and it is the abstracted essential feature of Simulated Annealing metaheuristics [21]. Also, reheating probability is included following Adapted simulated annealing metaheuristics [19]. Comparison value class specifies, which earlier best solution is used in making the selection. This follows Late-Acceptance Hill Climbing metaheuristics [7].

Control component provides high-level control mechanism. These include parallelization of restarts (restarts are independent of each other in the model so they can be easily parallelized). Experimental design setup is responsible for making statistically reliable comparisons between runs and hyper heuristics class for adjusting the parameters. Monitoring (Table 5.) sets whether search run information is provided for the user. It can be used to diagnose problems and for teaching.

4.3 Computing the baseline results

The system described above was used to compute the baseline results. First, simple hill-climbing with one or three restarts was run 64 times for each case. Each computation of classification accuracy within a run was validated with 50 repeated holdout validation. Then the mean and standard deviations of the best of 64 runs (i.e. highest classification accuracy achieved in a run) were computed. Secondly, for comparison global maximum classification accuracy was computed by exhaustive search for each case with 50 times repeated holdout validation. Note, that the number of combinations for each case was different. The number of iterations used in the heuristic searches was 10% of the ones used in the global evaluation. Lastly, for the worst case hybrid heuristic hyperparameter optimization was done by setting a grid of eight hyperparameter combinations and then computing the mean of best of runs for each hyperparameter combination.

5 Results

The case companies are presented in the first column in Table 6. and the number of globally evaluated combinations in the second column. The mean and standard deviations of the best of runs of baseline hill-climbing with one or three restarts are shown in the third column. Global maximum of evaluating all the combinations is shown in the fourth column. The fifth column shows the mean of the best of runs divided by the global optimum. It represents the goodness of the heuristics compared to the most effective but inefficient method. Lempesti and Papua cases differ starting from the third decimal place. Hyperparameter optimization with hybridization in the weakest cases resulted in 0.69 mean of best of runs making the comparative ratio 0.89.

As for execution time, the largest case Innolink (320 combinations evaluated with 50 times repeated holdout validation) took 6 hours on an Intel 1.6Ghz computer.

6 Discussion

This paper focused on a preprocessing combinations optimization problem: how good are heuristic search methods compared to exhaustive search given a 10%-time constraint? The main results are (Table 6.) that 5/6 cases achieved median 98% global optimum in 10% of time with simple hill-climbing with one of three restarts. The weakest performing case achieved 89% after hybrid heuristic hyperparameter optimization. It should be noted that the global maximum is most likely not achievable by any expert reasoning [11]. Preprocessing combinations search space was characterized and behaviour of the objective function (i.e. classification accuracy) was found to be discontinuous and erratic but showing patterns and regimes. There were several points near the global optimum in all of the cases (Figure 1.), which is attributed to be the main reason for high performance level of simple hill-climbing with restarts.

Table 6. Baseline results

Case	Combs	Mean best of runs	Global max	Ratio
Innolink	3240	0,77±0,02	0,80	0,96
Toyota	1080	0,89±0,05	0,90	0,99
3StepIt	108	0,67±0,03	0,73	0,92
M-Files	360	0,65±0,06	0,79	0,82
Papua	648	0,93±0,02	0,95	0,98
Lempesti	648	0,93±0,02	0,95	0,98

A software package with a component model (Table 4.) was created and made freely accessible to study the performance of single-state heuristics in finding near-optimal preprocessing combinations. The package supports heuristic hyperparameter optimization and hybridization of five basic heuristics (Table 3.). The main research implication is that the baseline results provide a sound benchmark level for more sophisticated data preprocessing automation systems such as [24]. For practice the implications are that it is possible to build applications that efficiently preprocess business performance measurement system data without significant loss of solution quality.

There are four main limitations in the results. First, the number of industry cases was limited to six and consequently statistical testing of the results (including standard deviation of the holdout round classification accuracies in global max and within a heuristic run) was not done leaving uncertainty as to what extent the results can be generalized. Secondly, all the cases were from a specific domain of business performance measurement systems and no cross-domain evaluation was conducted. Thirdly, all data sets were small in size and execution time was measured on a coarse scale. Thus evaluation of scalability requires further clarification. Fourthly, no analysis was done regarding the success and failure conditions of heuristics. This concerns specifically the difference between the five best performance and the one worst performance case. Further research is needed and in progress to build policy-based approaches to preprocessing. The aim is to reduce the preprocessing time from hours to minutes and to scale up from small to big data sets. Reinforcement learning [33] and Monte Carlo tree search [6] can learn and store state-action-values (i.e. data state, preprocessing actions, delayed classification accuracy) either as neural network or as asymmetric tree, and thus provide near-optimal preprocessing actions directly or with significantly smaller amount of searches.

ACKNOWLEDGEMENTS

Professor emeritus Pertti Järvinen. After sales director Jarmo Laamanen Toyota Material Handling Finland, managing director Pekka Vuorela Innolink Group, sales director Mika Karjalainen 3StepIt, senior director Mika Javanainen M-Files, managing director Olli Vaaranen Papua Merchandising and managing director Sirpa Kauppila Lempesti. Reviewers.

References

1. Baghel, M., Argawal, S. and Silakari, S.: Survey of Metaheuristic Algorithms for Combinatorial Optimization, International Journal of Computer Applications, Volume 58, No.19 (2012)
2. Bianchi, L., Dorigo, M., Cambardella, L., and Gutjahr W.: A survey on metaheuristics for stochastic combinatorial optimization, an international journal of Natural Computing, Volume 8 Issue 2, Pages 239 - 287 (2009)
3. Blum, C., Aguilera, M., Roli, A., and Sampels, M.: HybridMetaheuristics, An Emerging Approach to Optimization, Springer (2008)
4. Blum, C, Puchinger, J., Raild, G., and Roli, A.: Hybrid metaheuristics in combinatorial optimization: survey, Applied Soft Computing, Volume 11, Issue 6, Pages 4135-4151 (2011)
5. Breunig, M. M., Kriegel, H-P., Ng, R. T., and Sander, J.: LOF: Identifying Density-Based Local Outliers, Proc. ACM SIGMOD 2000 Int. Conf. On Management of Data, pp. 93-104 (2000)
6. Browne, C., Powley, E., Whitehouse, D., Lucas, S., Cowling, P., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S.: A Survey of Monte Carlo Tree Search Methods in IEEE Transactions on Computational Intelligence and AI in Games, Vol. 4 (2012)
7. Burke, E.K and Bykov, Y.: The Late Acceptance Hill-Climbing Heuristic, Technical Report CSM-192 (2012)
8. Chapman, P., Clinton, J., Kerber, R., Khabaza T., Reinartz T., Shearer, C., and Wirth R.: Crisp-Dm 1.0 step by step data mining guide, Crisp-DM consortium (2000)
9. Crone, S.F., Lessmann S., and Stahlbock, R.: The impact of preprocessing on data mining: an evaluation of classifier sensitivity in direct marketing, European Journal of Operational Research Vol. 173:3, pp. 781–800 (2005)
10. Denning, P., Comer, D., Gries, D., Mulder, M., Tucker, A., Turner., A., and Young, P.: Computing as a discipline. Communications of the ACM 32, pp. 9–23 (1989)
11. Engel, J., Gerretzen, J., Szymanka, E., Jansen. Jeroen J., Downey G., Blanchet, L. and Buydens L.: Breaking with trends in preprocessing, TrAC Trends in Analytical Chemistry, Volume 50, October 2013, Pages 96–106 (2013)
12. Fayyad, U., Piatetsky-Shapiro G., and Smyth, P.: The KDD Process for Extracting Useful Knowledge from Volumes of Data, Communications of the ACM, Vol. 39, No. 11, pp. 27-34, (1996)
13. Franco-Santos, M., Kennerley M., Micheli P., Martinez V., Mason S., Marr B., Gray D., and Neely A.: Towards a definition of a business performance measurement system, Internation Journal of Operation and Production Management, Vol 27:8, pp. 784-801 (2007)
14. Glover, F.: Future Paths for Integer Programming and Links to Artificial Intelligence. Computers and Operations Research 13 (5): 533–549 (1986)

15. Glover, F., and Kochenberger, G.: Handbook of metaheuristic. Kluwer Academic Publishers (2002)
16. Guyon, I., and Eliseff, A.: An Introduction to Variable and Feature Selection, The Journal of Machine Learning, Volume 3, 3/1/2003, Pages 1157-1182 (2003)
17. Han, J., Kamber M., and Pei J.: Data mining: Concepts and Techniques, Morgan Kaufmann, San Francisco (2012)
18. Hu, M-X., and Salvucci S.: A Study of Imputation Algorithms, Institure of Education, Science, NCES, U.S. Department of Education (1991)
19. Ingber, L.: Adaptive simulated annealing (ASA): Lessons learned, Control and Cybernetics,Vol. 25 No. 1,pp. 33–54, 1996 (1996)
20. Kaplan R. S., and Norton D.P.: The Balanced Scorecard – Measures that Drive Performance, Harvard Business Review, Vol 71:1, pp. 71-9 (1992)
21. Kirkpatrick, S., Gelatt Jr, C. D., and Vecchi, M. P.: Optimization by Simulated Annealing". Science 220 (4598): 671–680 (1983)
22. Kriegel H-P., Borgwardt, K.M., Kröger P., Pryakhin A., Schubert, M., and Zimek, A.: Future trends in data mining, Data Mining and Knowledge Discovery, Vol. 15:1, pp. 87–97 (2007)
23. Kuhn, M., and Johnson K.: Applied Predictive Modeling, Springer, New York, NY (2013)
24. Komer, B., Bergstra, J., and Eliasmit, C.: Hyperopt-Sklearn: Automatic Hyperparameter Configuration for Scikit-Learn. In Proc. of the 13th Python in Science Conference (2014)
25. Longadge, R., Dongre, S.S., and Malik L.: Class Imbalance Problem in Data Mining: Review, International Journal of Computer Science and Network,, Vol. 2, Issue 1 (2013)
26. Luke, S.: Essentials of Metaheuristics, Lulu (2010)
27. Parejo, J., Ruiz-Corte, A., Lozano, S., and Fernandez. P.: Metaheuristic optimization frameworks: a survey and benchmarking, Soft Computing, Volume 16 Issue 3, March 2012, Pages 527-561 (2012)
28. Perner, P.: Data Mining on Multimedia Data, Lecture Notes in Artificial Intelligence, Vol. 2558. Springer Verlag, Berlin Heidelberg New York (2002)
29. Pyle, D.: Data Preparation for Data Mining, Morgan Kauffman, San Francisco, CA (2003)
30. Raidl, G.: A unified view of hybrid metaheuristics, In Almeida, F., Aguilera, M., Blum, C., Vega, J., Perez, M., Roli, A., Samples, M.(2006) Hybrid Metaheuristics (2006)
31. R Development Core Team.: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2016)
32. Sadig S., Yeganeh, K., Induska M.: 20 Years of Data Quality Research: Themes, Trends and Synergies, ADC '11 Proceedings of the Twenty- Second Australasian Database Conference - Volume 115, Pages 153-162 (2011)
33. Sutton, S. and Barto, G.: Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA (1998)
34. Sörensen, K.: Metaheuristics – the Metaphor Exposed, University of Antwerp Operations Research Group ANT/OR (2012)
35. Vattulainen, M.: A method to improve the predictive power of a business performance measurement system by data preprocessing combinations: two cases in predictive classification of service sales volume from balanced data. In Ahmad Ghazawneh, Jacob Nørbjerg and Jan Pries- Heje(eds.) Proceedings of the 37th Information Systems Research Seminar in Scandinavia (IRIS 37), Ringsted, Denmark (2014)
36. Vattulainen, M.: metaheur R package, https://cran.r-project.org/web/packages/metaheur (2016)

37. Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G., Ng, A., Liu, B., Yu, P. S., Zhou, ZH., Steinbach, M., Hand, D.J., and Steinberg, D.: Top 10 algorithms in data mining, Knowledge and Information Systems, Vol. 14, Issue 1, pp. 1-37 (2008)
38. Wand Y., Wang R.: Anchoring Data Quality Dimensions in Ontological Foundations, Communications of the ACM, Vol. 39, No. 11 (1996)
39. Wu X., Zhu X., Wu, G-Q., and Ding W.: Data Mining with Big Data, IEEE Transactions on knowledge discovery and data engineering, Vol 26, No 1 (2013)
40. Yang Q., and Wu X.:. 10 Challenging Problems in Data Mining Research, International Journal of Information Technology and Decision Making, Vol. 5, No. 4, pp. 597–604 (2006)