A machine learning algorithm for identifying inconsistencies among sets of restrictions

**Keywords:** Machine learning, data validation, inconsistency detection, automation, statistical & economic models, data collection, data agnostic, synthetic data generation, optimisation, sets of restrictions, non-linear optimisation, neural networks, gradient algorithms

# Introduction

While there has been extensive literature in official statistics on data validation, there has been limited work done on mechanisms that would allow validating the validation rules. The present paper introduces an approach on identifying logical inconsistencies in complex economic or statistical sets of restrictions and generating synthetic data based on the non-cross violating sets of available constraints. The proposed implementation was developed as a part of an automated testing procedure of the newly introduced Centralised Submission Platform at the European Central Bank (ECB). This platform has been developed to address the need to collect data from diverse sources, such as economic, statistical, supervisory data, etc. and in various file formats. The platform supports the collection of multiple data types (e.g. strings, numbers, booleans etc.) and from several reporting agents (e.g. credit institutions, national central banks, research institutions). Upon files reception, data quality checks (both technical and business) are applied instantly to the underlying data in order to verify the quality of a data submission. Data producers at the ECB and the reporting agents are notified immediately in case of inconsistencies in the reported data. Reporting agents and users at the ECB are enabled to discuss, if needed, any failed business validations using a specific functionality of the platform and to reach an alignment. The platform offers a validation engine that allows the users to define their own business validation rules according to their needs, which may be relatively simply or represent an advanced statistical or economic model on the received data and to efficiently spot potential data quality issues. These checks are represented in the form of assertions in the platform’s validation engine.

In order to verify that the platform is functioning according to high business standards, we developed a machine learning program that (i) identifies logical and computational inconsistencies among various complex business validation checks on the incoming data and (ii) generates synthetic data satisfying the declared statistical or economic models. The program is able to find logical inconsistencies among business validation rules and identify which of them result in a cross violation to the model’s equilibrium. The algorithm is data agnostic, as it uses as input only the a-priori defined validation checks, regardless of any collection agreements or data specificities. It checks the structure and potential internal conflicts of the defined business validation models and finally generates synthetic data that are provided to the platform for performing additional technical tests of the validation engine. In case the constraints underlying the business validation models are not well defined and result in inconsistencies, subsets of these constraints are randomly selected by the program, until a system of restrictions with feasible solutions is found.

# Methods

The program treats the original problem as a non-linear optimisation problem, where the constraints are the business validation rules. These rules are analysed by a custom parser, via the execution of multiple regular expression pattern matching rules and then they are defined as mathematical equations in the form of user defined functions. There is a pre-defined list of function names and behaviour in the program that matches names and functionalities of the parsed validation rules. When the matching is finalised, the original business validation rules are interpreted by the program as user defined functions in the code, enabling the manipulation of the original constraints by the solving algorithms. A similar approach is followed for identifying the variables of the mathematical equation. The program gives the option to the user to define the objective function in advance, or to select one of the equations as the objective by random selection. Various optimisation methods from a wide range of the machine learning toolkit (e.g. Generalised Reduced Gradient algorithm, Nonlinear Conjugate Gradient method, Recurrent Neural Networks (RNN) and Artificial Neural Networks (ANN) – mainly for identifying potential temporal and computational gains compared to the known non machine learning approaches) can be applied to the algorithm. If the solving algorithm finds a feasible region for the system of constraints, then it generates the values that satisfy the constraints by assigning them to the respective variables. In case the system of equations is identified as infeasible by the solver, then the program uses subsets of the available constraints to find potential feasible solutions for each subset. The approach of Leave One Out Cross Validation (LOOCV) is used to determine which equation prevents the system from reaching an equilibrium. When all the possible combinations are tested, the program generates a report, informing the user about the feasibility of the initial set of constrains and all the feasible constraint combinations that can be derived from the original system. For the subsets that are able to generate feasible results, random data are generated and saved in the original form and file format of the received file. The synthetic data can be used to perform further tests on the platform’s validation engine, such as performance tests.

# Results

At a first level we verified the computation and running time of the used algorithms by simulating random data and functions that acted as the set of restrictions. Our initial findings suggested that the Generalised Reduced Gradient (GRG) algorithm is faster in terms of computation time when the set of restrictions were non-linear functions. This particular algorithm outperformed the others used in the tests, where the restrictions or objective functions were of exponential and quadratic form. On the other side, the Simplex algorithm is able to find solutions for feasible set of restrictions under a reasonable execution time. However, with the introduction of more complex restrictions, in terms of functional form, the algorithm was not able to find all the available feasible regions, in which the set of restrictions was in equilibrium. Apart from these checks, additional machine learning algorithms like Recurrent Neural Networks (RNN) and Artificial Neural Networks (ANN) were used. These algorithms were not as efficient as GRG or the Newton-Krylov algorithm, but we should point out that with the addition of restrictions, the score, in terms of computational efficiency was increased. That behaviour was expected, since these algorithms tend to be more efficient with large amounts of data as input [4]. All benchmarking tests were performed on the same hardware platform.

On the second level, we used both real restrictions coming from various data collections used in the ECB’s Centralised Submissions platform, along with many random generated restrictions that followed either an economic or statistical model. Our initial findings, regarding the optimization algorithms were verified with these tests, with some exceptions. The GRG algorithm, proved to be one of the most efficient algorithms, particularly with the real business restrictions, as it was able to generate data for all the feasible sets of restrictions that the program was able to identify. One important remark is that for data collections with few but overlapping validation rules, the execution time of this algorithm increased by approximately 10% to 15%. For the randomly generated restrictions, the results were similar. For quite complex restrictions, containing multiple variables in small validation rules sets, the computation time increased up to 45%, specifically in cases where the number of variables was significantly higher than the number of restrictions.

The Newton-Krylov algorithm ended up as the second best alternative to GRG, in terms of computational efficiency. For the same business test cases, it had similar results to the previous algorithm. For the artificially generated data, it had a slightly decreased efficiency in terms of computational time (up to 35% compared to GRG).

For the machine learning algorithms, the computational efficiency for the business validation rules was significantly lower compared to the previous algorithms. However, with the randomly generated data, after multiple iterations, they have matched or even surpassed the original algorithms. Even when using complicated sets of restrictions, they found feasible solutions, under a reasonable time period, even in cases where the previous algorithms failed.

During the execution of the various benchmarks and tests, the amount of used RAM did not show any statistically important fluctuations (the machine learning algorithms had the greater consumption). Regarding the CPU load, especially during the execution of complex computations, it fluctuated from 65% to 100%, for almost all algorithms, used.

# Conclusions

The program was able to identify inconsistencies for both the business oriented validation rules and the randomly generated ones, for almost all the cases and generate synthetic data for all subsets of feasible constraints combinations. These data were used for further testing the functionality of the ECB’s platform validation engine. From our research, we found that the usage of mathematical and machine learning related optimization algorithms can lead to effectively spot logical inconsistencies in sets of multiple restrictions, but the machine learning approach prevails on highly complicated sets of restrictions, in terms of finding a feasible solution within a short time. Apart from that, with the generation of synthetic data, we managed to test further the validation engine of the Centralised Submission Platform and verify that the pre-defined high quality operations standards are met.

# References

1. Peter N. Brown, Youcef Saad, Convergence Theory of Nonlinear Newton–Krylov Algorithms, SIAM Journal on Optimization (1994), 297–330.
2. Jameela Al Otaibi, Zeineb Safi, Abdelâali Hassaïne, Fahad Islam, Ali Jaoua, Machine Learning and Conceptual Reasoning for Inconsistency Detection, IEEE (2016), 338 - 346.
3. Behrooz Keshtegar, Mahmoud Miri, Introducing conjugate gradient optimization for modified HL-RF method, Engineering Computations Vol. 31 No. 4 (2014), 775-790.
4. M. Fazle Azeem, M. Hanmandlu, N. Ahmad, Structure identification of generalized adaptive neuro-fuzzy inference systems, IEEE Volume 11 Issue 5 (2003), 666 - 681.